

---

# **TECHNICKÁ UNIVERZITA V LIBERCI**

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: 2612R011 – Elektronické informační a řídicí systémy

## **Řízení modelu letadla pomocí PLC Mitsubishi**

### **The Control of Airplane model by PLC Mitsubishi**

#### **Bakalářská práce**

Autor:

**Jakub Nosek**

Vedoucí práce:

Ing. Lukáš Hubka, Ph.D.

Konzultant:

Ing. Petr Školník, Ph.D.

**V Liberci 10. 5. 2012**

# Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis

## **Poděkování**

Na tomto místě bych rád poděkoval vedoucímu mé bakalářské práce Ing. Lukášovi Hubkovi, Ph.D. Také bych rád poděkoval své rodině za trpělivost, kterou se mnou měli, během studia

## **Abstrakt**

Bakalářská práce se zabývá využitím programovatelného průmyslového kontroléru od firmy Mitsubishi Electric pro aplikaci - řízení letadla. Řízení, ovládané pomocí dotykového panelu - GOT 1000, nabízí dvě metody řízení. Byl použit algoritmus pro změnu parametru regulátoru. Ten umožňuje plynulé ovládání v nelineární části charakteristiky. Parametry regulátoru je možné zadat jak ručně, tak i programově.

Klíčová slova: PLC, řízení, automatizace

## **Abstract**

Bachelor work deals with the use of programmable industrial controller from Mitsubishi Electric. To apply: Control of the aircraft. The control algorithm provides two control methods controlled by touch panel, Algorithm was used to change the parameter controller. Controller parameters can be entered both manually and programmatically.

Keywords: PLC, control, automation

# Obsah

Prohlášení .....	2
Poděkování .....	3
Abstrakt .....	4
Seznam Ilustrací .....	7
Seznam Grafů .....	8
Seznam použitých termínů a zkratk .....	9
Úvod .....	10
1 Použité PLC .....	11
1.1 Struktura PLC systému .....	11
1.2 Použité moduly .....	13
1.2.1 Napájecí zdroj .....	13
1.2.2 Analogové moduly (převodníky A/D – D/A) .....	13
1.3 Dotykový panel GOT1000 .....	15
2 Zjištění statických i dynamických vlastností modelu .....	16
2.1 Statické vlastnosti .....	16
2.2 Dynamické vlastnosti .....	18
3 Návrh PID regulátoru a jeho implementace do PLC .....	23
3.1 Návrh regulátoru .....	23
3.2 Implementace do PLC .....	29
3.2.1 Vývojové prostředí .....	29
3.2.2 Řídicí algoritmus .....	31
3.2.3 Implementace .....	32
3.3 Porovnání simulace a reálné regulace .....	33
3.4 Vyhodnocení .....	34
4 PID s proměnným zesílením .....	36
4.1 Nelinearita .....	37
4.2 I a D koeficienty .....	37
4.3 Aplikace a měření .....	41
5 Grafický panel .....	44
5.1 Komunikace .....	44
5.2 Vytváření programu .....	45

6 Online monitoring a export dat do PC .....	48
Závěr.....	50
Seznam použité literatury .....	51
Příloha A – foto letadla .....	52

# Seznam Ilustrací

Obr. 1.1 - Struktura PLC.....	11
Obr. 1.2 – Multi-procesorové řízení .....	12
Obr. 1.3 - Popis řídící jednotky .....	12
Obr. 1.4 - Nastavení switchů pro D/A, viz [3] .....	14
Obr. 1.5 - Nastavení switchů pro A/D, viz [3] .....	14
Obr. 1.6 – Ukázka aplikace na panelu .....	15
Obr. 2.1 – Simulační schéma pro měření statických vlastností.....	16
Obr. 2.2 – Obecný graf přechodové odezvy .....	21
Obr. 3.1 – Schéma simulovaného regulátoru.....	28
Obr. 3.2 – Schéma simulované soustavy.....	29
Obr. 3.3 – Příklad nastavení komunikace v GX Developer .....	30
Obr. 3.4 – Nastavení Switchů v GX Developer .....	30
Obr. 3.5 – Ukázka přetypování.....	31
Obr. 3.6 – Kód PID regulátoru .....	31
Obr. 3.7 – Přenos v programu GX Developer .....	32
Obr. 4.1 – Schéma pro porovnání modulu a PLC .....	33
Obr. 5.1 – Změna P složky v programu.....	42
Obr. 6.1 – Zapojení RS 232 .....	44
Obr. 6.2 – Hlavní okno.....	45
Obr. 6.3 – Ukázka skriptu – Scatter Graph, resekce zobrazení grafu.....	46
Obr. 6.4 – Ukázka Base okna .....	47
Obr. 7.1 – Monitoring.....	48
Obr. 7.2 – Schéma pro záznam dat .....	49

# Seznam Grafů

Graf 1 - Měření statické charakteristiky.....	17
Graf 2 – Měření statické charakteristiky s nezávislými osami.....	17
Graf 3 – Statická charakteristika.....	18
Graf 4 - Měření dynamických vlastností.....	19
Graf 5 - Dynamické vlastnosti – buzení (červená), výstup systému (modrá).....	19
Graf 6 - Identifikace .....	20
Graf 7 - Tečna pro přenosovou funkci.....	22
Graf 8 – Proložení systémem 1. řádu .....	22
Graf 9 - Simulovaná regulace – vlastní nastavení (modrá barva), tabulkové (zelená).....	25
Graf 10 - PID s nastavení podle Cohena a Coona.....	25
Graf 11 – PID s nastavení podle Zieglera a Nicholse .....	27
Graf 12 – PID s nastavením podle Modifikace ZN pravidel.....	27
Graf 13 – Porovnání regulace .....	34
Graf 14 – Porovnání regulace .....	34
Graf 15 - Porovnání statických charakteristik .....	36
Graf 16 – Aproximovaná křivka vyjádřená rovnicí 4.1 .....	38
Graf 17 - Aproximovaná křivka vyjádřená rovnicí 4.2 .....	38
Graf 18 - Aproximovaná křivka vyjádřená rovnicí 4.3 .....	39
Graf 19 - Aproximovaná křivka vyjádřená rovnicí 4.3 .....	39
Graf 20 – Průběhy pro koeficienty z tab. 2.....	40
Graf 21 – Průběhy pro koeficienty $T_I=16.39$ a $T_D=0.92$ .....	41
Graf 22 – Výpočet P.....	42
Graf 23 – Záznam polohy bez algoritmu pro změnu P složky .....	43
Graf 24 – Záznam polohy s algoritmem pro změnu P složky .....	43



## Seznam použitých termínů a zkratek

PLC	Programmable Logic Controller, programovatelný logický automat
$T_v$	vzorkovací perioda [s]
$r_0$	zesílení regulátoru spojitého PID
$T_i$	integrační časová konstanta spojitého PID [s]
$T_d$	derivační časová konstanta spojitého PID [s]
USB	Universal Serial Bus, rozšiřující port počítače
RS 232	standardní komunikační rozhraní pro sériovou komunikaci mezi počítači
CPU	Central Processing Unit, procesor
$w$	žádaná veličina
$y$	regulační veličina
$u$	akční veličina
$e(t)$	regulační odchylka v časové oblasti
$e(k)$	regulační odchylka v číslicové regulaci
$T_u$	doba průtahu [s]
$T_n$	doba doběhu [s]

# Úvod

První zmínky o programovatelných průmyslových kontrolérech pocházejí ze Spojených států amerických, kde byly vyvinuty pro potřeby urychlení výrobních procesů v průmyslových továrnách. První PLC byla tvořena stovkami až tisíci dvoupolohových nebo vícepolohových relé a časovači a velmi těžko se přizpůsobovaly novým řídicím procesům, protože neměly možnost rozšiřujících modulů. PLC bylo vždy konstruováno pro konkrétní aplikaci.

Dnešní průmyslové kontroléry se rozdělují na dva typy – kompaktní a modulární. Výhodou kompaktních kontrolérů je, že všechno potřebné pro řízení aplikace je integrováno v jednom pouzdře, ale nevýhodou je právě nemožnost rozšíření o přídavné moduly. Zato u modulárních PLC je velkou výhodou jejich přizpůsobení pro danou aplikaci a vyměnitelnost přídavných modulů při jakékoliv nehodě.

Pro aplikaci, řízení modelu letadla, mi byl vybrán programovatelný průmyslový kontrolér Q25HCPU od firmy Mitsubishi Electric díky jeho lepším přizpůsobením pro danou aplikaci, rozšiřitelným modulům a převodníkům s proměnným rozsahem. Řízení celé úlohy bude grafickým dotykovým panelem. Kde bude zobrazována aktuální poloha a bude možné měnit žádanou hodnotu. Ovládání celé úlohy bylo vytvořeno za pomoci grafického dotykového panelu. Na něm lze měnit parametry regulátoru a žádanou akční hodnotu. Aktuální poloha letadla byla zobrazena v grafu na panelu.

PLC i grafický panel mi zapůjčila Technická univerzita v Liberci.

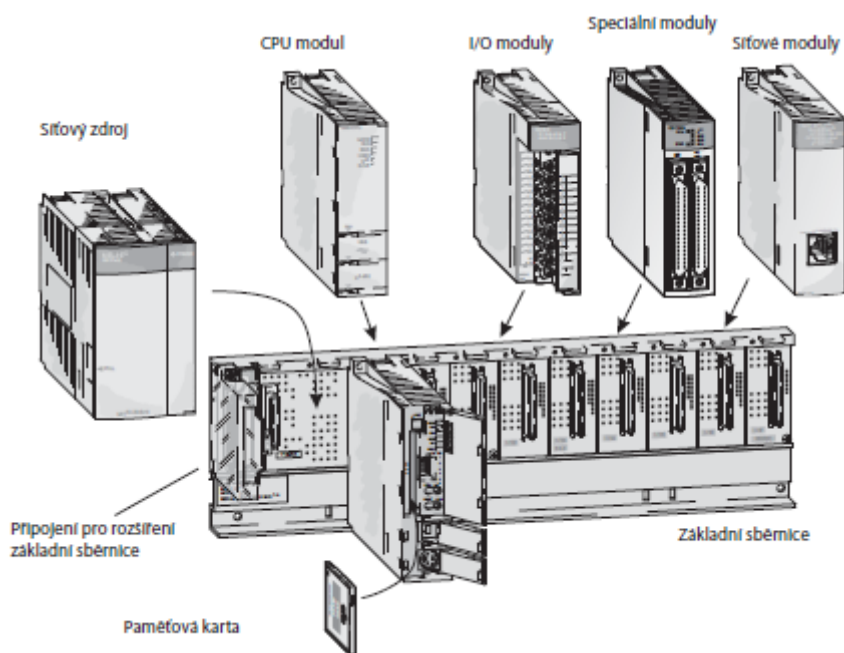
# 1 Použité PLC

Aplikované PLC vychází z řady *MELSEC Q*, konkrétní typ má označení Q25HCPU.

## 1.1 Struktura PLC systému

MELSEC systém Q je výkonné modulární PLC s multiprocessorovou technikou, tzn. využití více procesorů ve stejném čase. Slovo „modulární“ znamená, že konfiguraci PLC je možné individuálně a optimálně upravit podle dané aplikace. Při optimalizaci PLC pro danou úlohu jsem vyházel z manuálů, viz [2] a [3].

Jádro PLC je tvořeno sběrnici, na kterou je možné připojit síťový zdroj, minimálně jednu řídicí jednotku CPU, rozšiřitelné moduly – A/D a D/A převodníky, digitální vstupy a výstupy, síťový modul a speciální moduly.



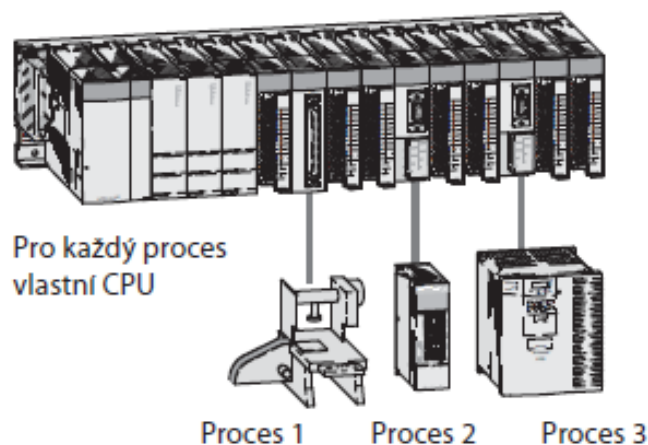
Obr. 1.1 - Struktura PLC

Komunikace mezi jednotlivými moduly a řídicí jednotkou probíhá přes interní sběrnici. Sběrnice, na které je umístěna řídicí jednotka CPU, je označována jako základní sběrnice. MELSEC nabízí 5 různých základních sběrnic až s 12 zásuvnými sloty pro moduly.

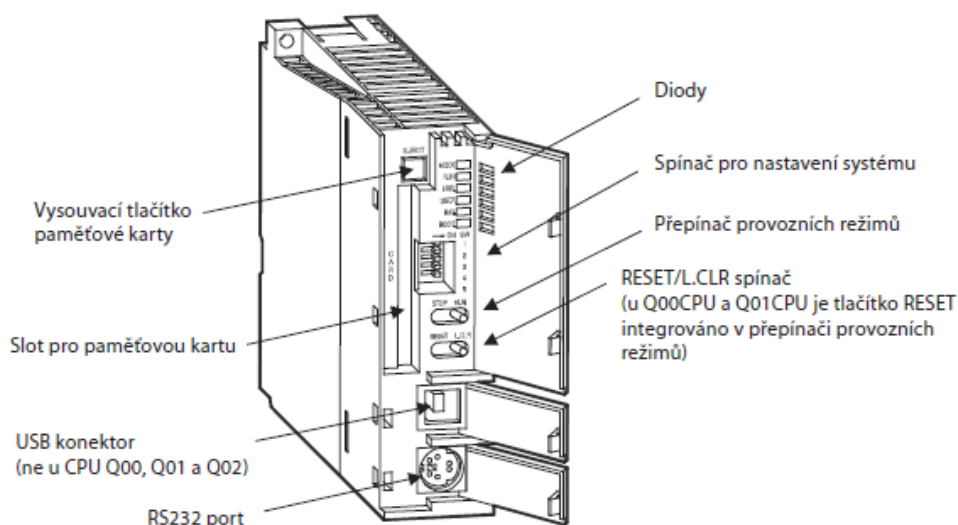
Každou základní sběrnici je možno doplnit o rozšiřující sběrnice, díky kterým jsou k dispozici další zásuvné sloty. Sběrnice jsou mezi sebou propojeny pomocí připojovacích kabelů. Tyto kabely slouží pro napájení instalovaných modulů. Základní sběrnici je možné rozšířit až o 7 rozšiřujících sběrnic. Na základní a na rozšiřujících sběrnicích je možno instalovat až 64 modulů.

Při výběru napájecího zdroje je třeba zohlednit příkon vstupních a výstupních modulů, speciálních modulů a periferních zařízení. V případě potřeby musí být použita rozšiřující sběrnice s dalším síťovým zdrojem.

U složitého systému, kde není výkon jednoho řídicího CPU dostatečný, je možno úkoly rozdělit na více CPU a tím výkon systému zvýšit viz obr 1.2 – Multiprocessorové řízení.



Obr. 1.2 – Multi-procesorové řízení



Obr. 1.3 - Popis řídicí jednotky

## **1.2 Použité moduly**

Rozšiřující moduly, které jsem použil pro účely regulace modelu letadla, byly A/D a D/A převodníky a napájecí zdroj.

### **1.2.1 Napájecí zdroj**




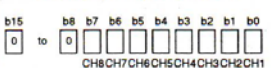

Zdroj pracuje se stejnosměrným napětím 5 V. K dispozici jsou i síťové zdroje s napětím 24 V DC nebo 100 až 240 V AC. Výstupní napětí přímo napájí sběrnici a není dostupné na svorkách. V úloze jsme použili napájecí zdroj Q63P, který splňoval požadavky na napájení řídicí jednotky CPU a rozšiřujících modulů.

### **1.2.2 Analogové moduly (převodníky A/D – D/A)**




Při řízení reálného systému není většinou možné použít diskrétní hodnoty. Je potřeba použít převodníky, které nám buď z analogových hodnot převádí na diskrétní hodnoty, to jsou analog/digitální (A/D) převodníky. Nebo opačný princip D/A převodníky.

Při řešení reálných aplikací je potřeba vybrat vhodné moduly, které se mohou od sebe odlišovat rozlišením a počtem analogových vstupů či výstupů. Rozlišení udává, jaká nejmenší hodnota může být analogovým modulem zaznamenána, respektive jaký je kvantizační krok. Rozlišení je určeno strukturou analogového modulu a závisí na tom, kolik bitů je zapotřebí k reprezentaci dané veličiny. Dalším parametrem je počet vstupů/výstupů analogového modulu, které jsou také označovány jako kanály.

Velkou výhodou analogových modulů, použitých v mé úloze, je velice rychlý převod hodnot, udávaná rychlost je 80  $\mu$ s, viz [2]. Vesměs všechny dostupné analogové moduly pracují se stejným jmenovitým rozsahem napětí a proudů. Existují analogové moduly pouze pro zpracování proudových veličin, napěťových veličin nebo takové, které mohou zpracovávat obě veličiny. Analogové moduly mají i nastavitelný rozsah, který se nastavuje pomocí tzv. switchů a jejich nastavení je popsáno níže na obrázcích.

Setting item	
Switch 1	Output range setting (CH1 to CH4) 
Switch 2	Output range setting (CH5 to CH8) 
Switch 3	<div> <div>           For Q62DAN and Q64DAN   </div> <div>           HOLD/CLEAR function setting            0H : CLEAR            1 to FH (numeric value other than 0H)* : HOLD         </div> </div> <div> <div>           For Q68DAVN and Q68DAIN            b15 to b0   </div> <div>           HOLD/CLEAR function setting            0 : CLEAR            1 : HOLD         </div> </div>
Switch 4	 00H : Normal mode (non-synchronized) 01 to FFH (numeric value other than 00H)* : Synchronized output mode 0H : Normal resolution mode 1 to FH (numeric value other than 0H)* : High resolution mode 0H : Normal mode (D/A conversion processing) 1 to FH (numeric value other than 0H)* : Offset/gain setting mode
Switch 5	0 : Fixed

Obr. 1.4 - Nastavení switchů pro D/A, viz [3]

Setting	
Switch 1	Input range setting 
Switch 2	Input range setting 
Switch 3	Not used
Switch 4	 00H : With temperature drift correction 01 to FFH (numeric value other than 00H)* : Without temperature drift correction 0H : Normal resolution mode 1 to FH (numeric value other than 0H)* : High resolution mode 0H : Normal mode (A/D conversion processing) 1 to FH (numeric value other than 0H)* : Offset/gain setting mode
Switch 5	0 : Fixed

Obr. 1.5 - Nastavení switchů pro A/D, viz [3]

Bližší nastavení příslušných switchů je popsáno v kapitole 3.2.1 Vývojové prostředí. Pro reálný proces jsem využil analogových modulů pod typovým označením Q64AD (A/D převodník) a Q62DA (D/A převodník). Tyto převodníky lze označit za univerzální, protože nezpracovávají výhradně jeden typ měřené veličiny.

## 1.3 Dotykový panel GOT1000

Dotykový panel GOT1000 usnadňuje práci jak programátorům, tak i pracovníkům údržby a obsluhy. Díky 64 bitovému procesoru je chod grafických aplikací velmi rychlý a reakce na dotyk nejsou zpožděné. Díky resistivní dotykové vrstvě lze ovládat panel i v rukavicích.

Panel je vybaven komunikačním rozhraním Q Bus - sběrnice rozhraním, díky kterému komunikuje se základní deskou, na níž je umístěn procesor a rozšiřující moduly. Pro komunikaci s PC je panel vybaven dvěma porty: RS 232, tak i USB, pomocí nichž je přenos dat velmi snadný i rychlý.



*Obr. 1.6 – Ukázka aplikace na panelu*

Bližší popis nastavení komunikace a programování v programu GT Designer 2 ve vlastní kapitole 5 Grafický panel.

## 2 Zjištění statických i dynamických vlastností modelu

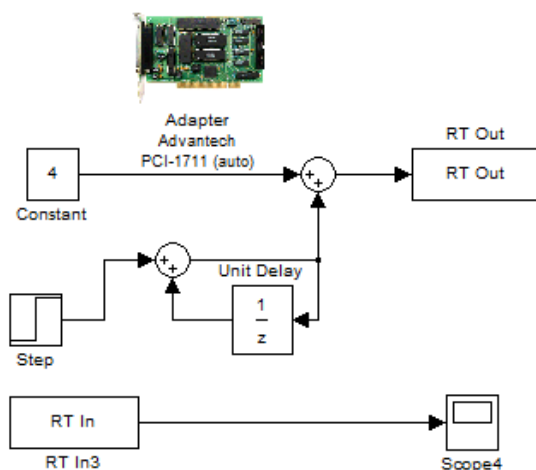
Pro zjištění převodní charakteristiky je potřeba zvolit pracovní oblast, či pracovní bod, v kterém se dále zjišťují dynamické vlastnosti modelu. Z těchto vlastností se určí převodní charakteristika modelu ve zvolené pracovní oblasti

### 2.1 Statické vlastnosti

Pro určení pracovní oblasti je potřeba určit statickou charakteristiku modelu, závislost výstupního napětí na vstupním napětí. Rovněž je nutné zvolit přírůstek napětí na vstupu modelu. Já jsem zvolil krok po 0.25 V. Také je zapotřebí určit dobu ustálení, po odzkoušení jsem ji určil na 120 s.

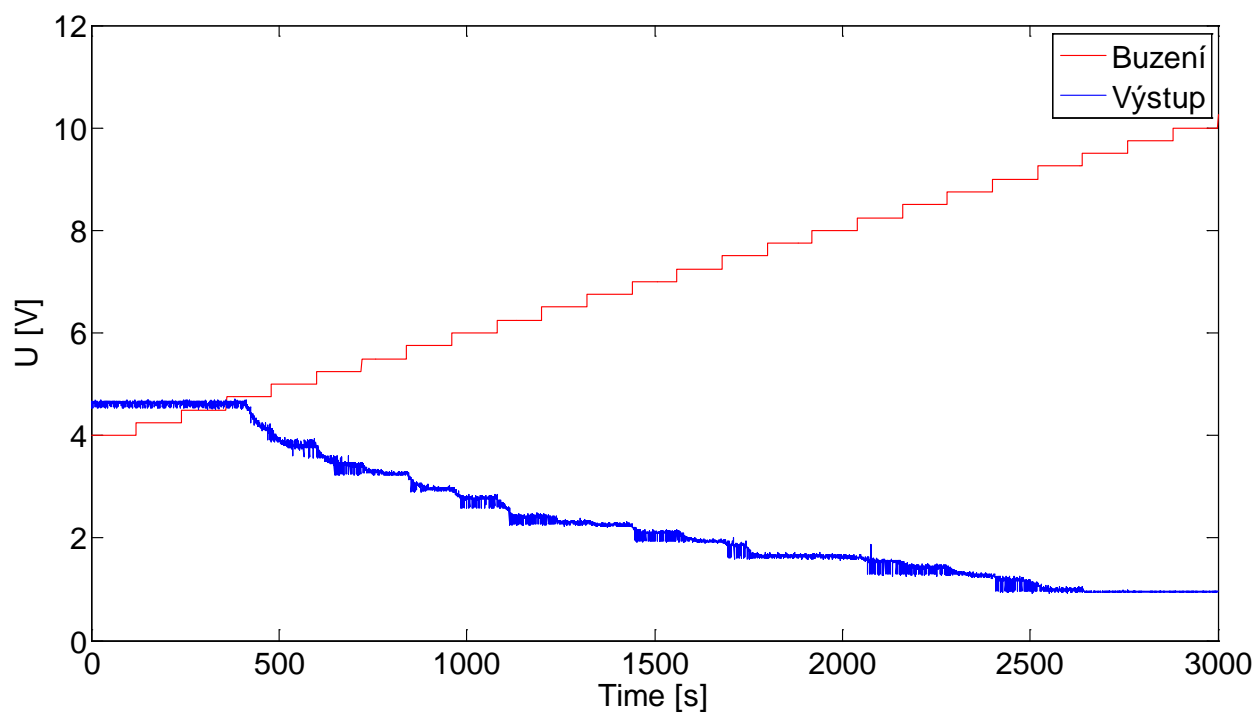
Pro zpracování jsem použil program *MATLAB* a multifunkční měřicí kartu v PC, která obsahuje D/A a A/D převodníky. Vzorkovací periodu jsem nastavil na 0.25 s. Vstupní napětí jsem volil od 4 V do 10 V. Při velikosti vstupního napětí 4 V nebyl motor letadla schopný se roztočit.

Ve schématu, obr. 2.1, jsem použil Unit Delay spolu se sumátorem pro zvýšení hodnoty o 0.25 V každých 120 s. Měření začínalo na 4 V, kvůli mrtvé zóně motoru.

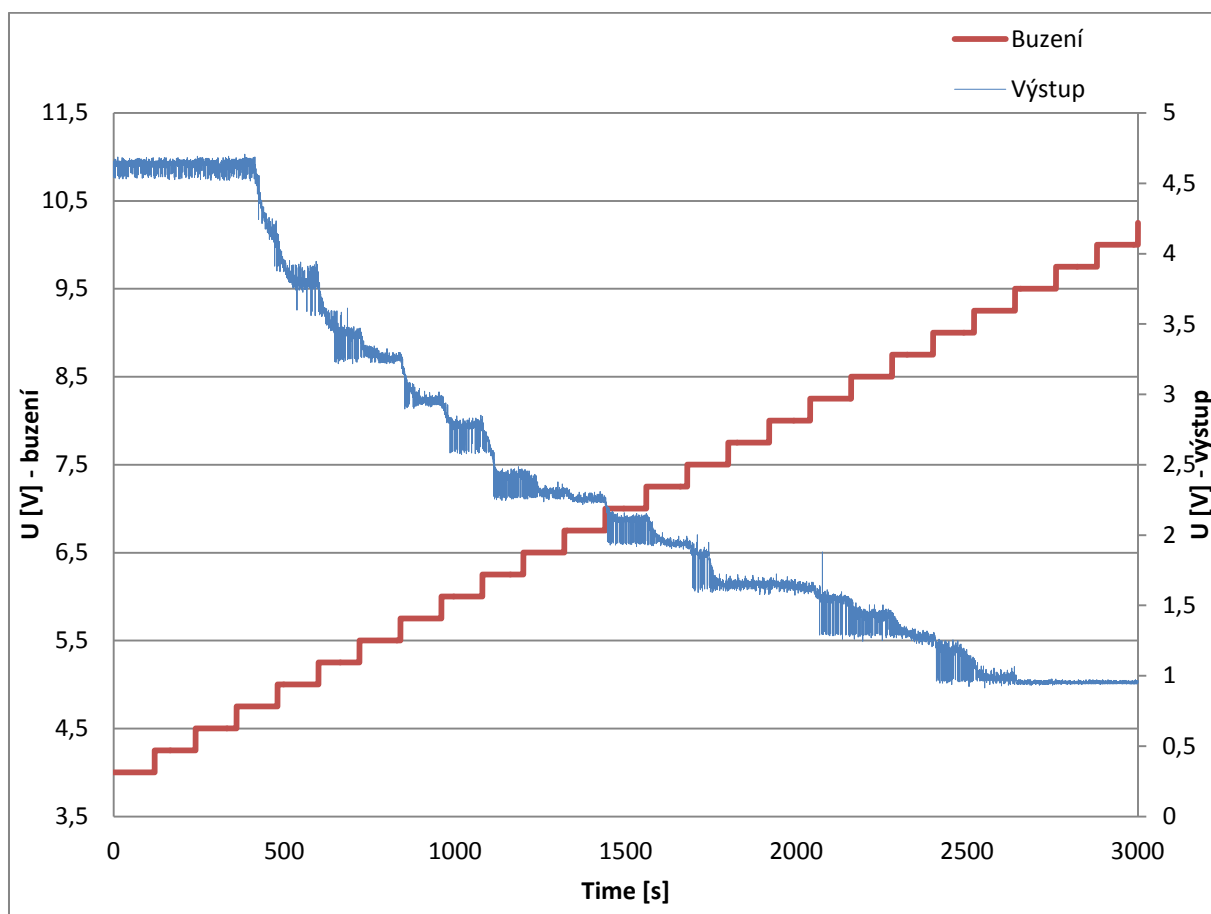


Obr. 2.1 – Simulační schéma pro měření statických vlastností



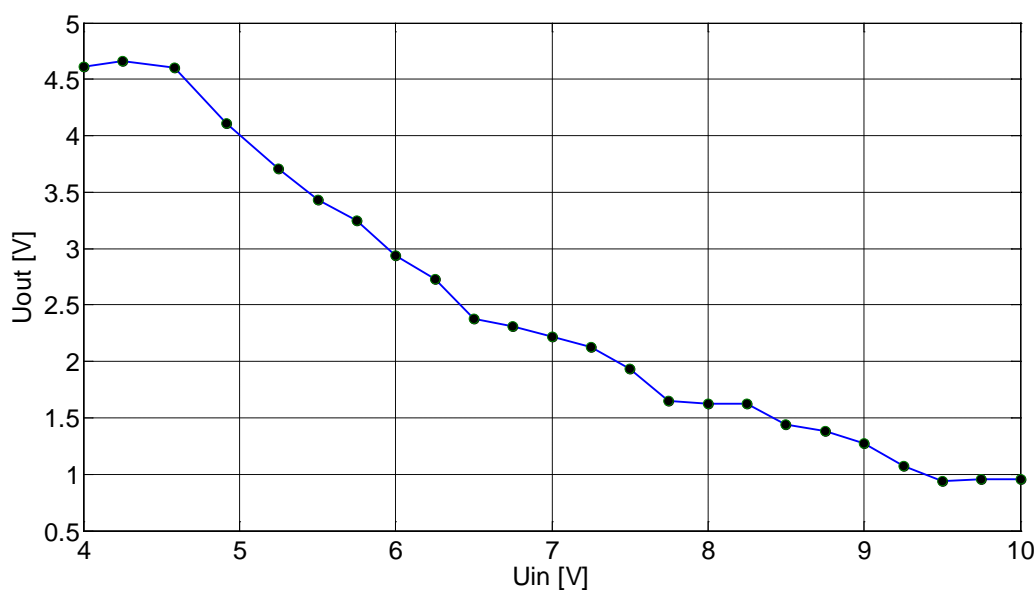


Graf 1 - Měření statické charakteristiky



Graf 2 – Měření statické charakteristiky s nezávislými osami

Graf 2 je tvořen v programu MS Excel, z důvodu rozdílného rozlišení dvou nezávislých os. Po úpravě naměřených dat jsem dostal následující charakteristiku.



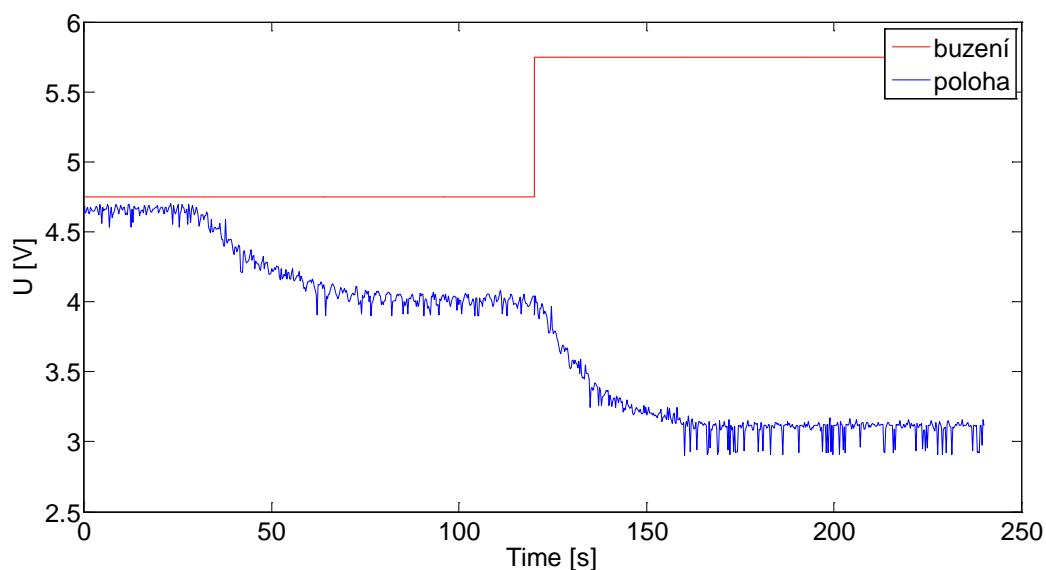
*Graf 3 – Statická charakteristika*

Pracovní oblast jsem při následující identifikaci, která vychází ze statické charakteristiky, určil od 4.5 V do 6 V vstupního napětí. Statické zesílení v pracovní oblasti je -0.748. V této oblasti je možné proložit statickou charakteristiku přímkou.

## 2.2 Dynamické vlastnosti

Po určení statických vlastností a pracovní oblasti jsem se mohl pustit do dalšího kroku, identifikace systému z pohledu dynamických vlastností.

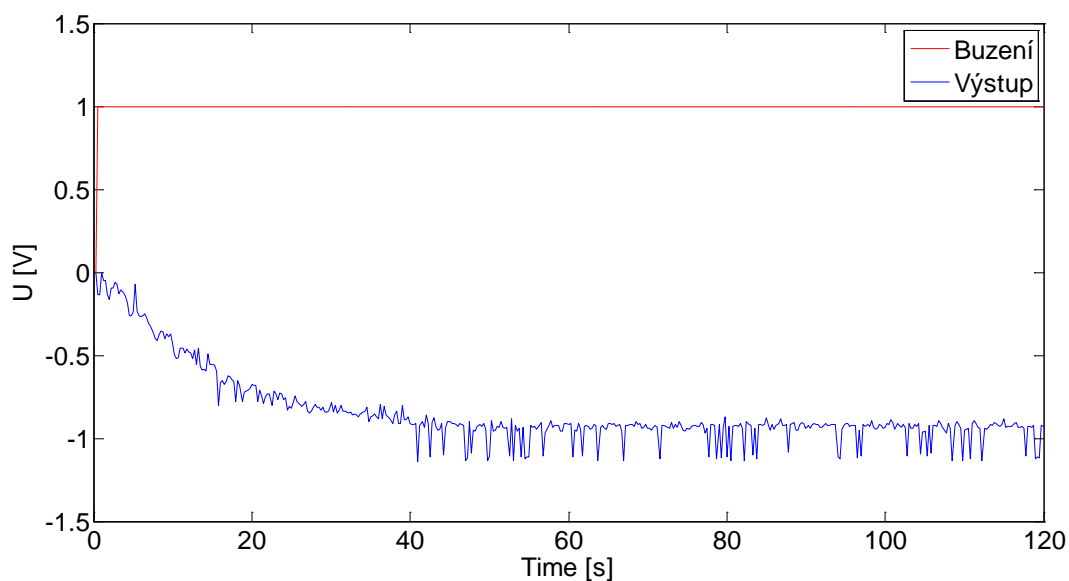
Letadlo bylo nutné vybudit skokovou změnou vstupního napětí z klidové polohy, ale nesměla být překročena pracovní oblast. Opět pomocí programu *MATLAB* jsem zaznamenával celé měření, z ustálené hodnoty opět do ustálené.



*Graf 4 - Měření dynamických vlastností*

Prodleva v prvních třiceti sekundách měření, viz graf 4 je způsobená startem letadla. Sice se začalo již pohybovat, ale ještě nenabralo dostatečnou rychlost. Až dosáhne potřebné rychlosti, bude již vztlak dostatečný a letadlo se zvedne. Tuto situaci si můžeme představit jako start reálného letadla.

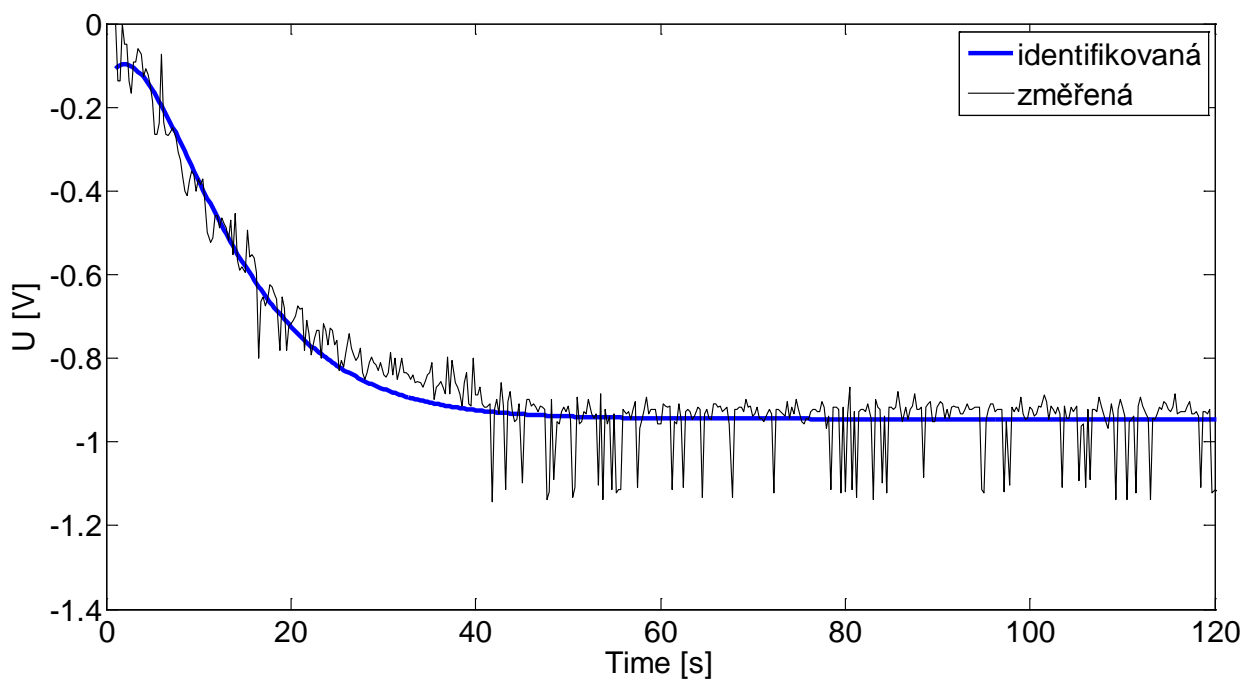
Naměřené hodnoty se museli upravit – ořezat. Důležité hodnoty pro mé měření byly až od skokové změny. Dále se musel naměřený průběh posunout, aby hodnoty začínaly na nulové hodnotě, viz graf 4.



*Graf 5 - Dynamické vlastnosti – buzení (červená), výstup systému (modrá)*

Po naměření a upravení dynamických vlastností modelu, v určené pracovní oblasti, následovala identifikace. Z upravených dat jsem za pomoci funkce `ident` v programu MATLAB identifikovat převodní funkci.

Identification toolbox je jednoduchý nástroj programu MATLAB, který slouží pro nalezení přenosu soustavy. Umožňuje nám vybrat řád přenosu, dopravní zpoždění, určit zda kořeny budou reálné nebo komplexní a také omezit intervaly pro hledání parametrů.



Graf 6 - Identifikace

Po nalezení příslušných koeficientů vyšla přenosová funkce (2.1) druhého řádu se dvěma reálnými kořeny ( $x_1 = -0.1466$ ,  $x_2 = -0.1464$ ) a zesílením  $K_p = -0.9451$ . Funkce druhého řádu odpovídá naměřeným datům, kde je malá doba prodlevy, která u funkce prvního řádu není. Oba kořeny charakteristické rovnice leží v levé polorovině Gaussovy roviny, což je důkazem, že je systém stabilní a aperiodický.

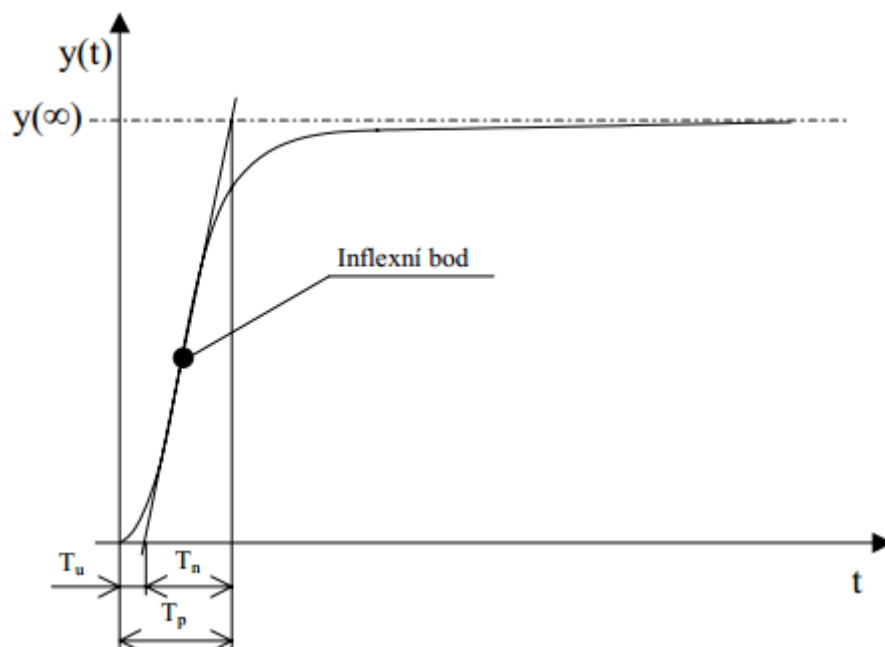
$$F(s) = \frac{-0.9451}{46.5765s^2 + 13.6494s + 1} \quad (2.1)$$

Modrá křivka (Graf 6) odpovídá 71.25 % shodě s naměřenými hodnotami. Vyšší přesnost by byla možná, pouze kdyby naměřený signál nebyl zatížen šumem, viz graf 4.

Pro zjištění doby průtahu a doběhu bylo potřeba simulovat přechodovou odezvu systému. Využil jsem skript, který spočítá pomocí derivace pozici inflexního bodu na

přechodové charakteristice. V inflexním bodě vytvoří tečnu k charakteristice. Hodnota doby průtahu je průsečíkem tečny s osou  $x$  (obr 2.2). Viz [8]

Doba  $T_p$  ( $T_u+T_n$ ) vyjadřuje dobu, za kterou by odezva dosáhla ustálené hodnoty při zachování počáteční rychlosti změny.



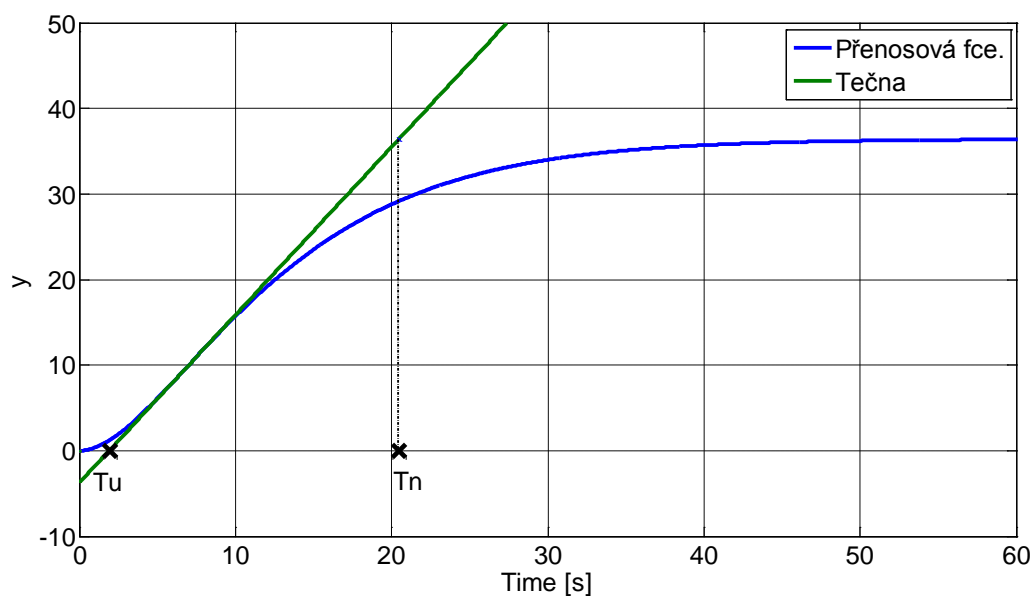
Obr. 2.2 – Obecný graf přechodové odezvy

Použitý skript pro výpočet časových konstant

```
jm = [46.5765 13.6494 1]
sys = tf(1,jm);
n = 10000; tmax = 50;
t = linspace(0,tmax,n);
T = tmax/(n-1);
y = step(sys,t);
dy = diff(y)/T;
[k i] = max(dy);
ti=t(i);
tecna = k*(t-ti)+y(i);
tec=tecna-y(i);
pom = 0; Tu=0;
for i=1:(n-1)
    if (tecna(i) >= 0 && pom == 0)
        Tu = t(i);
        pom=pom+1;
    end

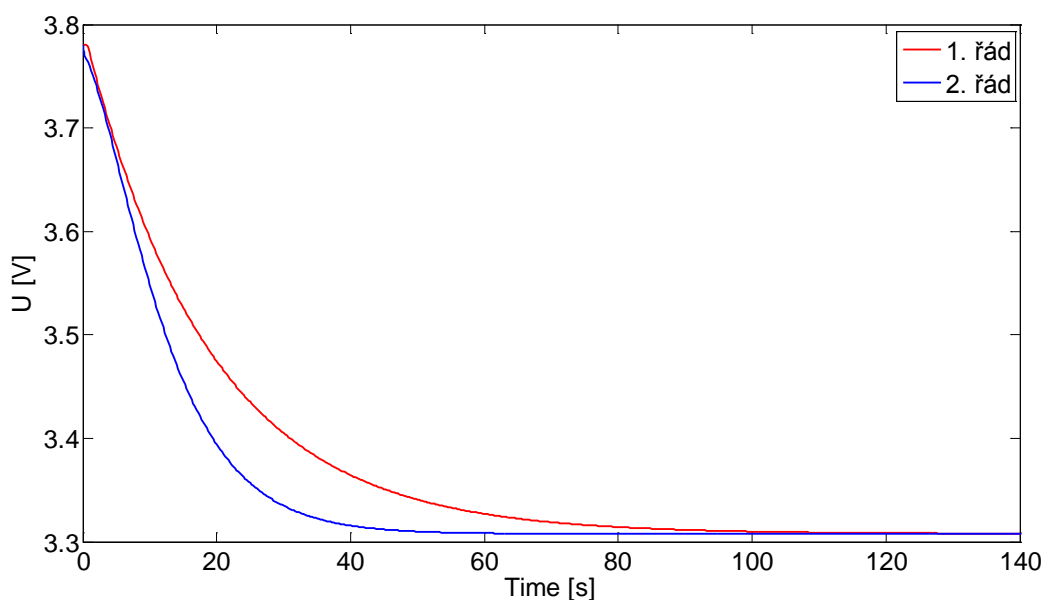
    if (tecna(i) >= y(n))
        Tx = t(i);
        break;
    end
end
Tn = Tx-Tu;
```

Vypočtené časové konstanty:  **$T_u = 1.95$  s**    **$T_n = 18.54$  s**



Graf 7 - Tečna pro přenosovou funkci

Pro kontrolu správnosti výpočtu časových konstant, jsem porovnal systém druhého řádu popsaný rovnicí (2.1), z kterého jsem vycházel při výpočtu časových konstant. Tyto konstanty jsem vložil do systému prvního řádu s dopravním zpožděním (2.2). Oba systémy jsem graficky porovnal. Z grafu 8 je zřejmé, že výpočty byly provedeny správně.



Graf 8 – Proložení systémem 1. řádu

$$F(s) = \frac{-0.9451}{18.54s+1} e^{-s1.95} \quad (2.2)$$

## 3 Návrh PID regulátoru a jeho implementace do PLC

### 3.1 Návrh regulátoru

Při návrhu PID regulátoru jsem vycházel z návrhů popsaných v Prostředkách automatizačního řízení, viz [1]. Obsahují několik rozdílných návrhových metod. Obecně platí, že PID regulátory jsou popsány následující rovnicí v časové oblasti:

$$u(t) = r_0 \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (3.1)$$

K jeho číslicové realizaci je potřeba zvolit vhodnou numerickou náhradu integrálu a derivace. Pro aproximace integrační složky lze použít tzv. lichoběžníkovou metodu s rovnicí:

$$\frac{1}{T_i} \int_0^t e(\tau) d\tau \approx \frac{T_v}{T_i} \sum_{i=1}^k e(i) \quad (3.2)$$

Průběh regulační odchylky jako funkce času je nahrazen nikoliv po částech konstantní funkcí, ale lineárně rostoucími nebo klesajícími úseky. Používat přesnější numerický výpočet je zbytečné, neboť konstanta  $T_i$ , která to celé v převrácené hodnotě násobí, je odhadnuta na základě modelu.

Nejjednodušší aproximací derivační složky regulátoru je zpětná diference, kde od sebe odečítáme aktuální hodnotu výchylky s předchozí hodnotou.

$$T_d \frac{de(t)}{dt} \approx \frac{T_d(e(k) - e(k-1))}{T_v} \quad (3.3)$$

Kombinací rovnic popsaných výše dostaneme obecnou rovnici diskrétního PID regulátoru s okamžitou hodnotou akčního zásahu:

$$u(k) = r_0 \left[ e(k) + \frac{T_v}{2T_i} \sum_{i=1}^k (e(i) + e(i-1)) + \frac{T_d}{T_v} (e(k) - e(k-1)) \right] \quad (3.3)$$

Druhou možností je přírůstková metoda, která udává pouze přírůstek akčního zásahu a ne jeho aktuální hodnotu:

$$\Delta u = r_0 \left\{ e(k) \left[ 1 + \frac{T_v}{2T_i} + \frac{T_d}{T_v} \right] + e(k-1) \left[ \frac{T_v}{2T_i} - 1 - \frac{2T_d}{T_v} \right] + e(k-2) \frac{T_d}{T_v} \right\} \quad (3.4)$$

$T_v$  v popsaných rovnicích označuje periodu vzorkování. Přírůstkový tvar je vhodnější pro regulaci z hlediska výpočetního výkonu použitého PLC. Důvodem je to, že není zapotřebí v každém kroku znovu vypočítávat hodnotu akční veličiny, ale stačí pouze vypočítat její přírůstek od předchozího kroku a tento přírůstek přičíst k předchozí hodnotě akční veličiny.

Pro zjištění parametrů regulátoru jsem použil návrhovou metodu. Potřebné časové konstanty, pro výpočet parametrů regulátoru, jsem vypočítal pomocí skriptu v MATLABU, který je vypočítán z přenosové funkce systému.

*Tabulka 1 - Nastavení regulátoru podle Chiena, Hronese a Reswicka*

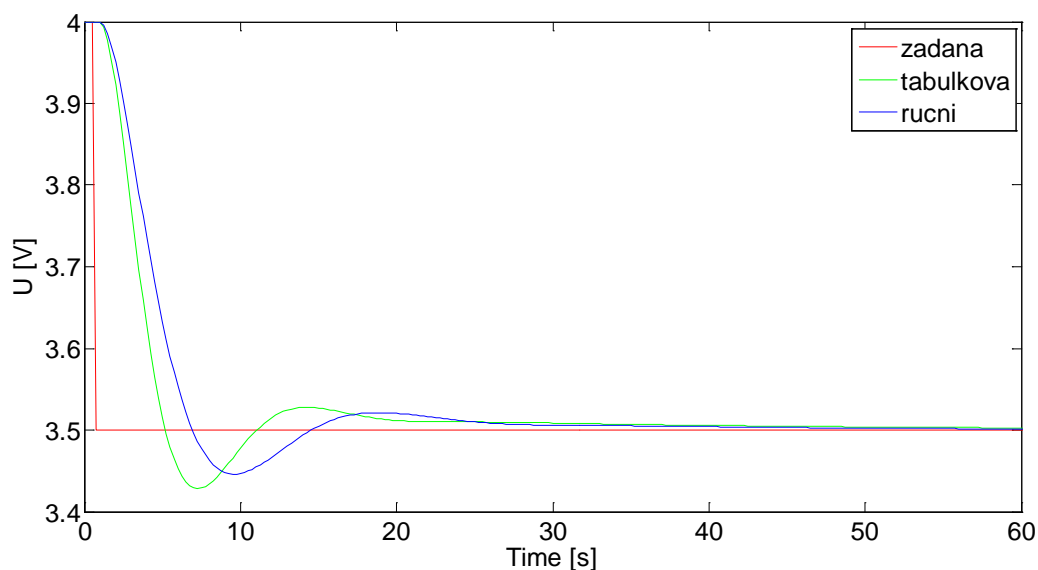
Regulátor	Aperiodický regulační pochod		Překmit 20%	
	Žádaná hodnota	Porucha	Žádaná hodnota	Porucha
P	$r_o=0,3 \tau(KT_D)$	$r_o=0,3 \tau(KT_D)$	$r_o=0,7 \tau(KT_D)$	$r_o=0,7 \tau(KT_D)$
PI	$r_o=0,35 \tau(KT_D)$ $T_i=1,2 \tau$	$r_o=0,6 \tau(KT_D)$ $T_i=4T_D$	$r_o=0,6 \tau(KT_D)$ $T_i=\tau$	$r_o=0,7 \tau(KT_D)$ $T_i=2,3T_D$
PID	$r_o=0,6 \tau(KT_D)$ $T_i=\tau$ $T_d=0,5T_D$	$r_o=0,95 \tau(KT_D)$ $T_i=2,4T_D$ $T_d=0,42T_D$	$r_o=0,95 \tau(KT_D)$ $T_i=1,35 \tau$ $T_d=0,47T_D$	$r_o=1,2 \tau(KT_D)$ $T_i=2T_D$ $T_d=0,42T_D$

Pro přepočet je potřeba  $T_D = T_u$  a  $\tau = T_n$

Po dosazení hodnot do tabulky pro aperiodický regulační obvod, zaměřený na regulaci žádané hodnoty, vyšli tyto koeficienty:  $r_0 = 9.56$ ,  $T_i = 25.029$  a  $T_D = 0.9165$  - znázorněné zelenou barvou. Nakonec jsem použil ručně nastavené hodnoty, které vycházejí z předešlých hodnot:  **$r_0 = 6.03$** ,  **$T_i = 18.54$**  a  **$T_D = 0.975$**  - znázorněné červenou barvou. Tyto hodnoty se v praktické zkoušce chovali mnohem lépe, i když mají delší dobu náběhu. Při simulování jiných nastavení PID regulátorů jsem vycházel z popisů nastavení v [1].

U metody nastavení regulátoru podle Chiena, Hronese a Reswicka jsem očekával rychlou dobu náběhu a dobrou regulaci i pro malé rozdíly žádané hodnoty. Obě očekávání se splnily.

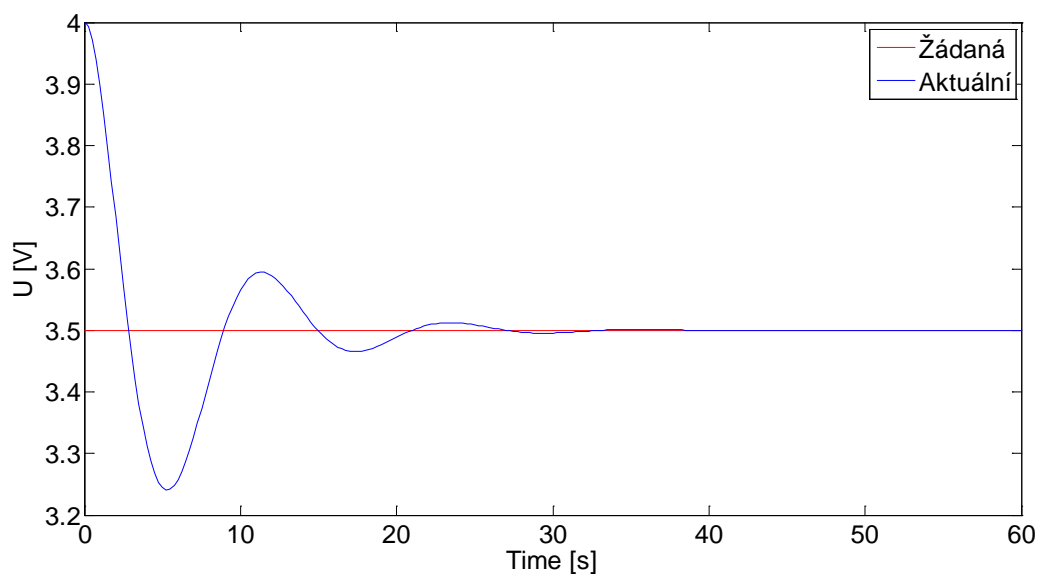




Graf 9 - Simulovaná regulace – vlastní nastavení (modrá barva), tabulkové (zelená)

Experimentálně, pomocí simulací, jsem zkoušel i jiné metody nastavení regulátoru, viz graf 10, 11 a 12. Měření bylo prováděno pouze pomocí simulací.

Metoda nastavení podle Cohena a Coona vylepšuje regulaci při větším normalizovaném dopravním zpoždění. Berou ho v potaz při výpočtech časových konstant regulátoru. Pro systémy s malým dopravním zpožděním má tato metoda velmi podobnou odezvu jako metodika návrhu podle pravidel Zieglera a Nicholse, což se v grafu 10 ukázalo.



Graf 10 - PID s nastavení podle Cohena a Coona

Tabulka 2 – Nastavení PID podle Cohena a Coona

Regulátor	$r_o$	$T_i$	$T_d$
P	$\frac{1}{K} (0,35 + \frac{1}{\Theta})$		
PI	$\frac{1}{K} (0,083 + \frac{0,9}{\Theta})$	$\frac{3,3 + 0,31\Theta}{1 + 2,2\Theta} T_u$	
PD	$\frac{1}{K} (0,16 + \frac{1,24}{\Theta})$		$\frac{0,27 - 0,88\Theta}{1 + 0,13\Theta} T_u$
PID	$\frac{1}{K} (0,25 + \frac{1,35}{\Theta})$	$\frac{2,5 + 0,46\Theta}{1 + 0,61\Theta} T_u$	$\frac{0,37}{1 + 0,19\Theta} T_u$

Normalizovaná zpoždění v tabulce 2 spočítáme jako poměr doby průtahu ku době doběhu.

Pravidla návrhu podle Zieglera a Nicholse, viz graf 11, se nespécializují pro regulaci skokových změn žádané hodnoty, ale na odstranění chyby v regulačním obvodu. Jak jsem psal výše, u systému s malým dopravním zpožděním, mají velmi podobné nastavení koeficientů jako metoda Cohena a Coona, tím pádem mají i velmi podobnou regulaci. Odchylka u nich je velmi malá. Ani jednu z těchto metod nelze použít. Obě mají velmi velký překmit a dlouho dobu ustálení. V dnešní době se považují za velmi dobrý první odhad pro nastavení regulátoru.

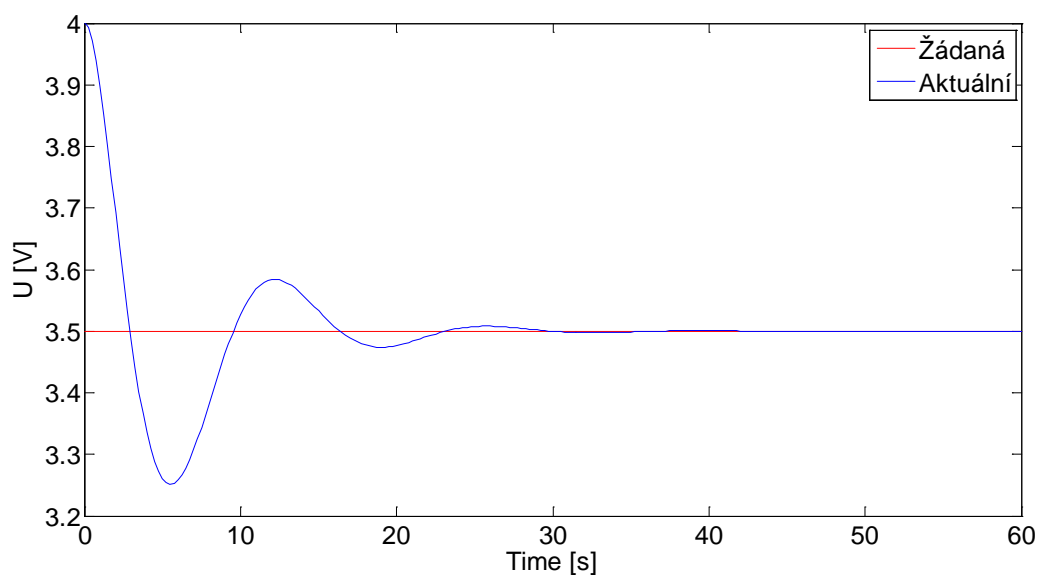
Tabulka 3 - Nastavení podle Zieglera a Nicholse

Regulátor	$r_o$	$T_i$	$T_d$
P	$1/(K \Theta)$		
PI	$0,9/(K \Theta)$	$3T_u$	
PID	$1,2/(K \Theta)$	$2 T_u$	$0,5 T_u$

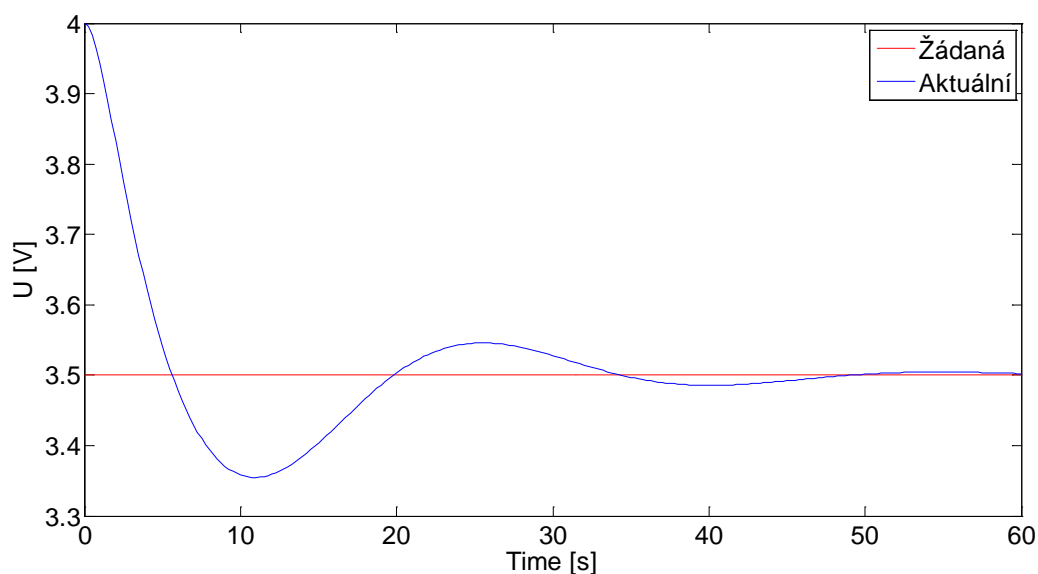
V grafu 11 jsem využil modifikaci pravidel Zieglera a Nicholse. Tato modifikace by měla zlepšit odezvu na změnu žádané hodnoty. To se projevilo, ale ani modifikací jsem se nedostal lepší odezvu jak u metody návrhu podle Chien, Hrones a Reswicka.

Tabulka 4 - Modifikace ZN pravidel

Přechodová odezva	$r_o$	$T_i$	$T_d$
s malým překmitem	$0,33r_k$	$0,5T_k$	$0,33T_k$
bez překmitu	$0,2r_k$	$0,5T_k$	$0,33T_k$



Graf 11 – PID s nastavením podle Zieglera a Nicholse



Graf 12 – PID s nastavením podle Modifikace ZN pravidel

Po několika pokusech, kde letadlo bylo řízeno z *MATLABU* za pomoci multifunkční karty jsem zjistil, že dosavadní struktura derivační složky regulátoru mi šum zesiluje a regulaci značně ztěžuje. Proto jsem se rozhodl pro jinou strukturu derivační složky PID regulátoru. Rovnice upravené derivační složky:

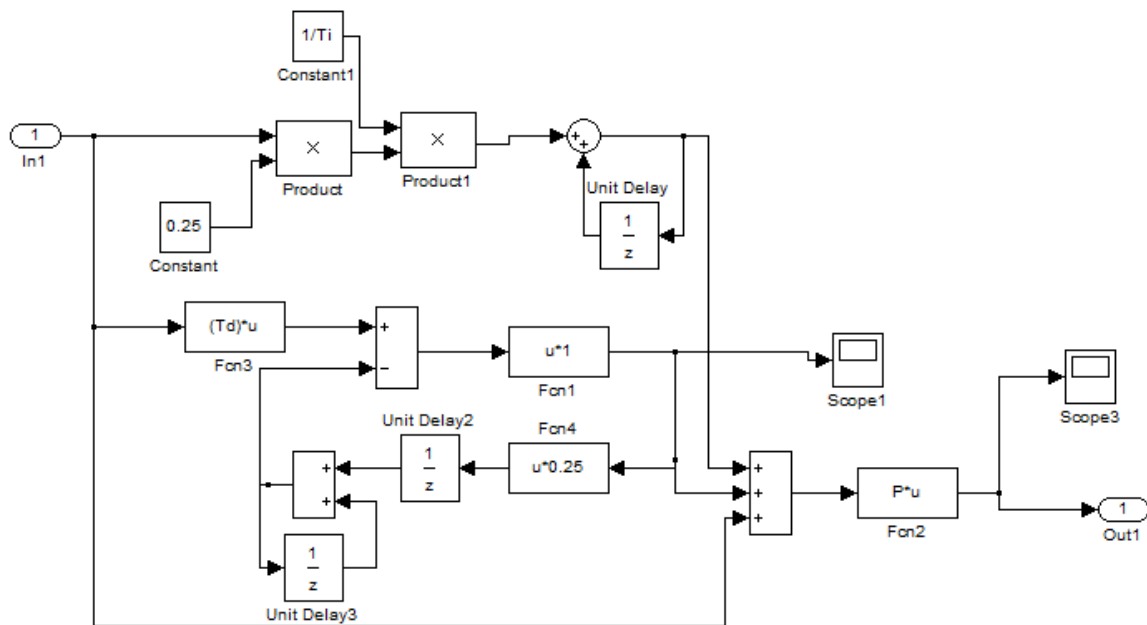
$$D(k) = T_D e(k) - T_v [D(k-1) + D(k-2)] \quad (3.5)$$

Podobný tvar derivační složky najdeme v číslicovém PID regulátoru v programu *MATLAB* ve verzi R2012a. Její popis lze najít v dokumentaci [10].

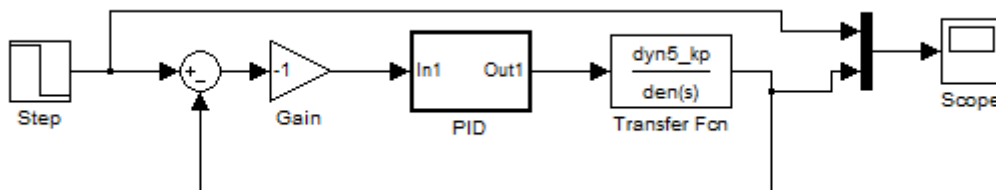
Rovnice použitého PID regulátoru:

$$u(k) = r_0 \left[ e(k) + \frac{T_v}{2T_i} \sum_{i=1}^k [e(i) + e(i-1)] + T_D e(k) - T_v [D(k-1) + D(k-2)] \right] \quad (3.6)$$

Pro lepší simulaci jsem vytvořil subsystém, jenž reprezentuje číslicový PID regulátor, který byl implementován do PLC.



Obr. 3.1 – Schéma simulovaného regulátoru



Obr. 3.2 – Schéma simulované soustavy

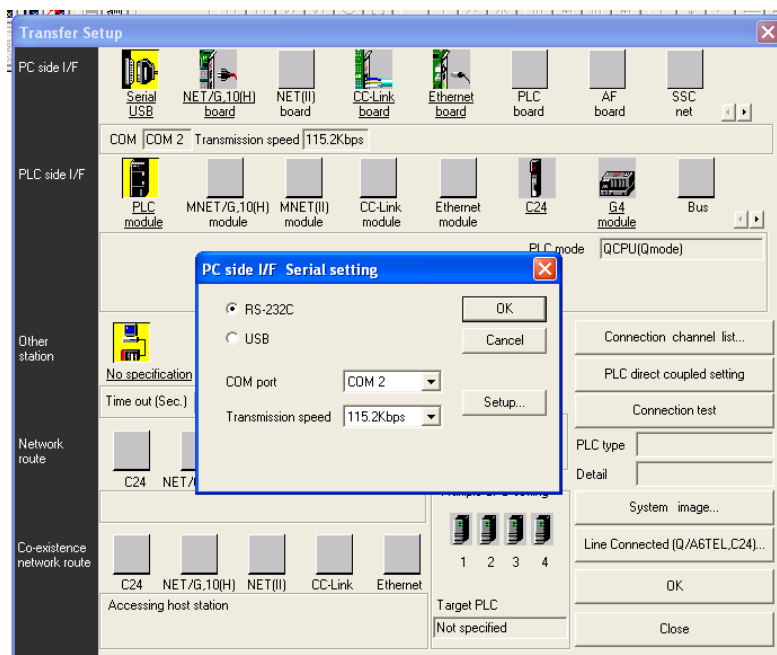
Aby byl dodržen regulační postup, bylo nutné invertovat výchylku. Důvodem je záporné zesílení systému (2.1), které způsobuje s rostoucím akčním zásahem, klesající hodnotu na výstupu.

## 3.2 Implementace do PLC

### 3.2.1 Vývojové prostředí

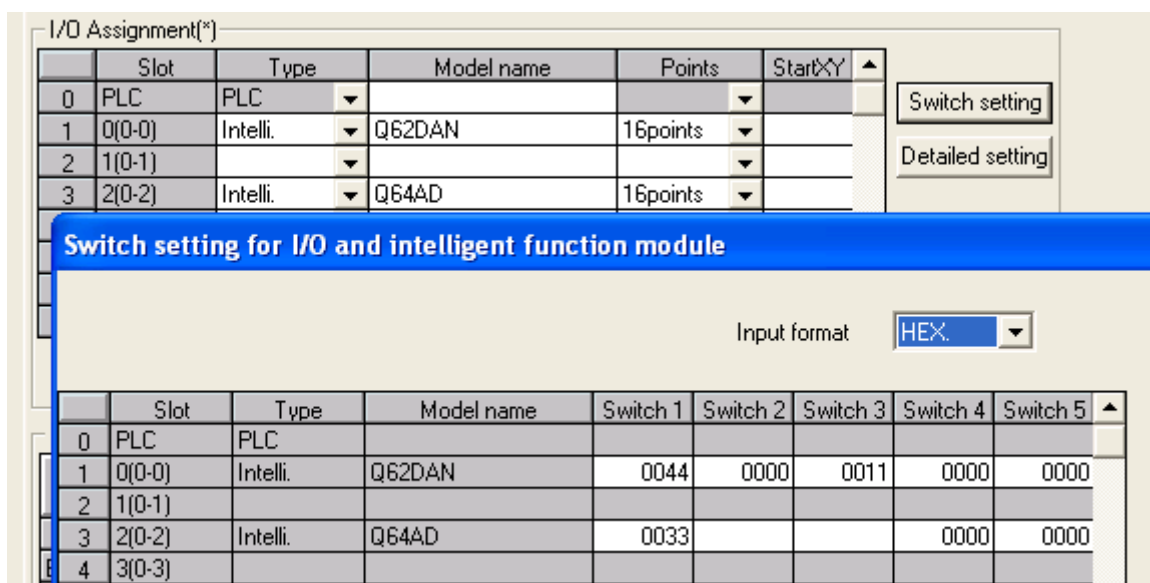
Pro programování výpočetních algoritmů pro řídicí jednotku jsem využil softwaru GX Developer Verze 8.62Q, který umožňuje vytvářet algoritmy ve třech programovacích strukturách: ladder, bloková schémata a strukturovaný text. Tento způsob programování jsem si zvolil. Takto jsem programoval jednotku v rámci bakalářského projektu, tudíž je to pro mě nejbližší způsob programování.

Komunikaci mezi CPU jednotkou a PLC lze zvolit více způsoby. Jedním způsobem je rozšiřující modul s ethernetovou přípojkou, který nám umožní programovat PLC na dálku, po síti. Řídicí jednotka Q25HC obsahuje dva porty pro komunikaci: USB a RS 232. Při práci na bakalářském projektu minulý rok, jsem neměl k dispozici potřebný software pro komunikaci po USB, tak jsem zvolil druhou možnost, a to RS 232. Po kladných zkušenostech z minulého roku jsem se rozhodl pro komunikaci po sériové lince i tento rok. Nastavení sériové komunikace jsem zvolil: 115.2 kbps s lichým paritním bitem.



Obr. 3.3 – Příklad nastavení komunikace v GX Developer

Po instalaci programu je nutné doinstalovat přídatné utility pro ovládání a nastavení rozšiřujících modulů, v tomto případě A/D a D/A převodníků. Tento software se nachází na CD-ROMU s programem GX Developer, pouze je potřeba otevřít složku Utilit a vybrat, co je třeba. Poté můžeme nastavit převodníky v programu. Nastavení se nachází v PLC parametrech v záložce I/O assignment. Pomocí switchů nastavíme rozsah převodníků viz [3]. Např. hodnota u Q62DAN switch 1 0044 znamená, že první a druhý převodník jsem nastavil do rozsahu  $-10 \div 10$  V. Bližší popis nastavení switchů lze najít v manuálech (viz. [3]).



Obr. 3.4 – Nastavení switchů v GX Developer

V programu jsou hodnoty převodníků vždy reprezentovány hodnotami od -4000, což odpovídá -10 V, až do 4000 (+10 V). Já jsem tyto hodnoty násobil 0.00125, abych se dostal zpět na rozmezí  $-10 \div 10$  V a hodnoty v programu byly lépe srozumitelné. Ještě bylo potřeba hodnoty přeměnit na typ real. O to se starala už integrovaná funkce v programu (obr. 3.5). Seznam všech interních funkcí v programu lze dohledat v příručce pro GX developer, viz [5].

```
reg_y:=(INT_TO_REAL(D21))*0.00125;
reg_e:=reg_w-reg_y;
reg_e:=reg_e*-1.0;
```

Obr. 3.5 – Ukázka přetypování

### 3.2.2 Řídicí algoritmus

```
(*I*)
reg_yI:=(reg_I*reg_e)*ts)+reg_zI;
IF (reg_yI > 1.66) THEN
    reg_yI:=1.66;
END_IF;
IF (reg_yI < 0.35) THEN
    reg_yI:=0.35;
END_IF;
reg_zI:=reg_yI;

(*D*)
reg_xD:=reg_zD+reg_YD;
reg_yD:=(reg_e*reg_D)-reg_xD;
reg_zD:=reg_yD*ts;
reg_YD:=reg_xD;

reg_u:=reg_P*(reg_e+reg_yD+reg_yI);

IF (reg_u > 10.0) THEN (*ošetření-saturace*)
    reg_u:=10.0;
END_IF;
IF (reg_u < 0.0) THEN
    reg_u:=0.0;
END_IF;

D12:=REAL_TO_INT(reg_u/0.0025);
D11:=REAL_TO_INT(reg_w/0.0025);
```

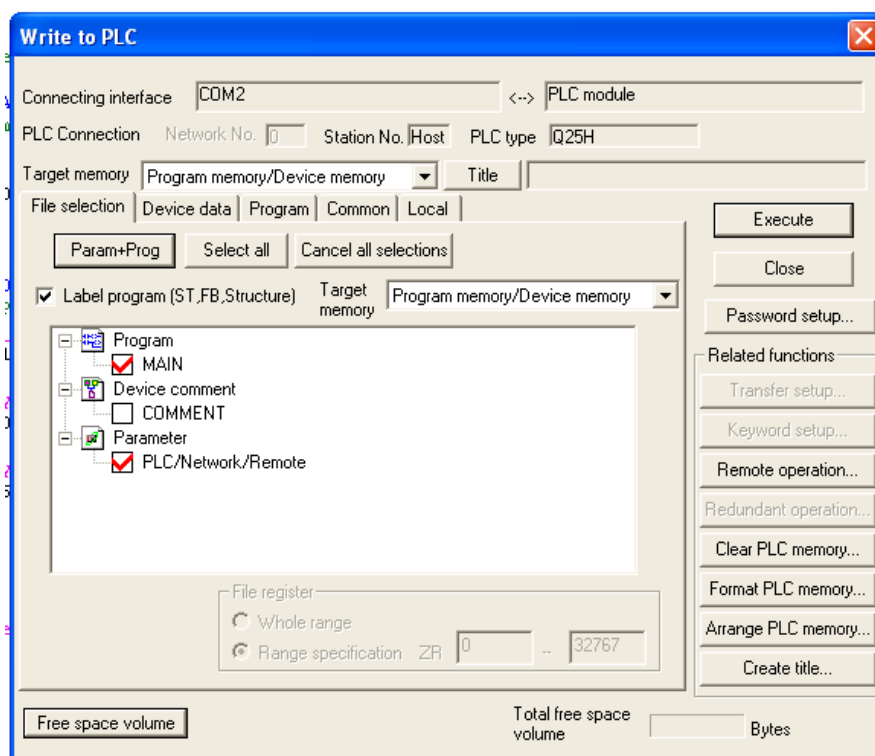
Obr. 3.6 – Kód PID regulátoru

V algoritmu jsem přidal dvojí ošetření, viz Obr. 3.6. Integrační složku omezují od 0.35, to z důvodu urychlení náběhu motoru. Když zanedbáváme derivační složku v klidu, tak při hodnotě integrační složky regulátoru 0.35, je výstupní napětí rovno 2.11 V. Při tomto

napětí se motor netočí. Toto omezení nezasahuje negativně do regulace. Opačně, když bude hodnota integrační složky 1.66, tak výstupní napětí má hodnotu přes 10 V a v tomto okamžiku se musí aplikovat druhé ošetření, které omezí hodnotu na výstupu na 10 V. To zabraňuje tomu, aby se do převodníku nedostala hodnota přesahující 4000, což by způsobilo chybu v programu a jeho ukončení.

### 3.2.3 Implementace

Pro nahrání řídicích algoritmů do CPU jednotky je potřeba program přeložit. Funkci lze najít v záložce Convert, dále Convert/Compile. Tento způsob lze provést klávesovou zkratkou - F4. Po úspěšné kompilaci můžeme program nahrát do PLC. Záložka Online a Write to PLC. Aby se povolila tato funkce, je potřeba mít správně navázanou komunikaci, která byla popsána o kapitolu výše.



Obr. 3.7 – Přenos v programu GX Developer

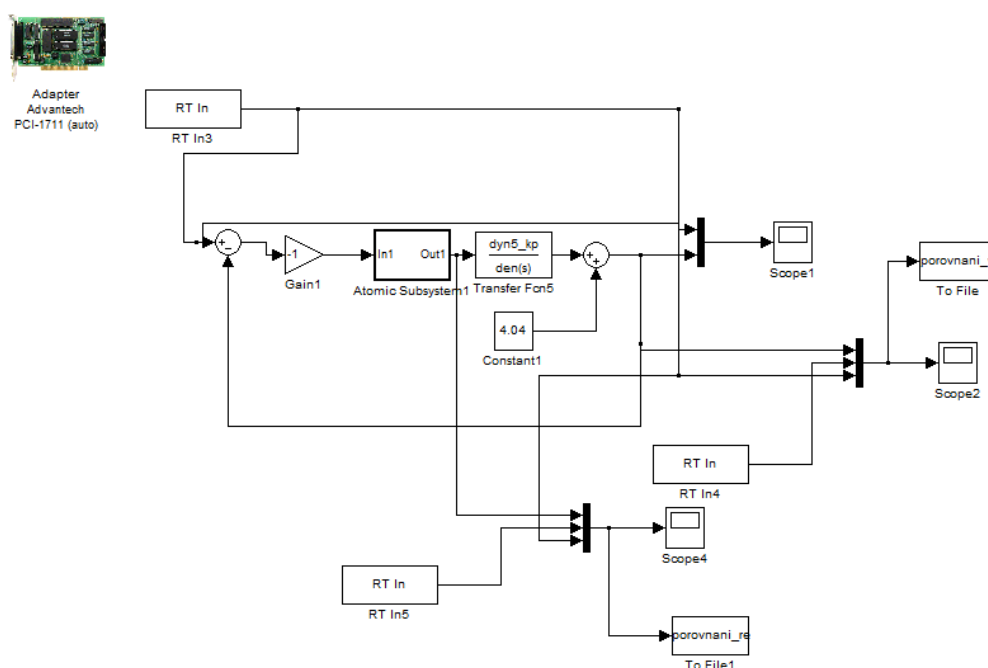
Při správně nastavené komunikaci se zobrazí přenosové okno. Zde si můžeme vybrat, co chceme přenést do PLC. Já používal zrychlenou volbu Param+Prog viz Obr. 3.8. Což označí Parametry a Program. Pro přenos je potřeba kliknout na Execute. Předtím je ovšem



lepší zastavit PLC pomocí přepínače na CPU jednotce. Když tak neprovedeme, tak se nás program zeptá, jestli chceme program zastavit. Dochází k velkému zpoždění a přenos trvá déle.

### 3.3 Porovnání simulace a reálné regulace

Po kompletním přenosu programu do PLC, jsem mohl PLC připojit k vstupům a výstupům na modelu a provést porovnání simulace s reálnou regulací pomocí PLC. Pro tento případ jsem si vytvořil speciální model v programu *MATLAB*.

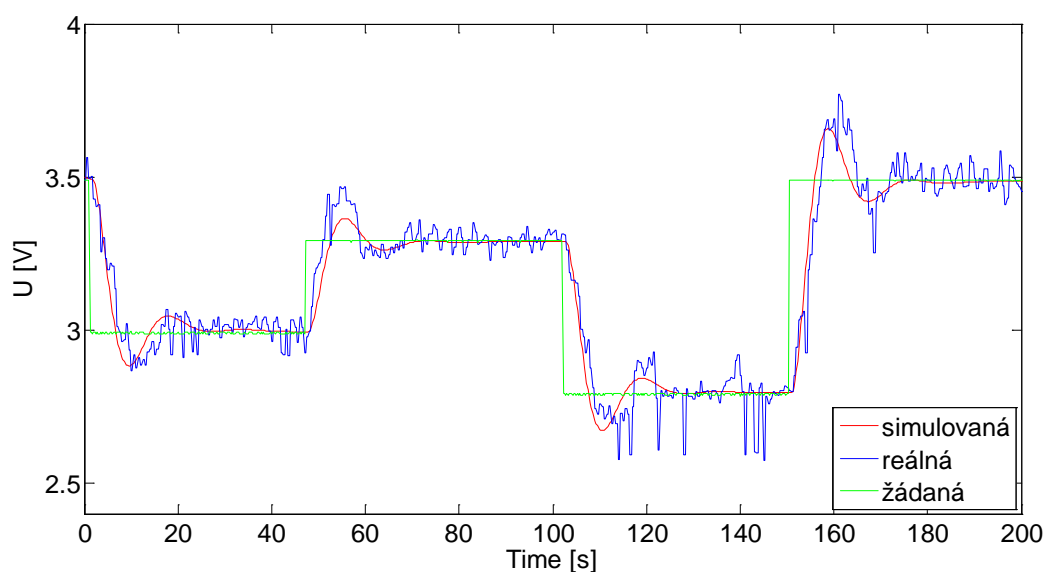


Obr. 4.8 – Schéma pro porovnání modulu a PLC

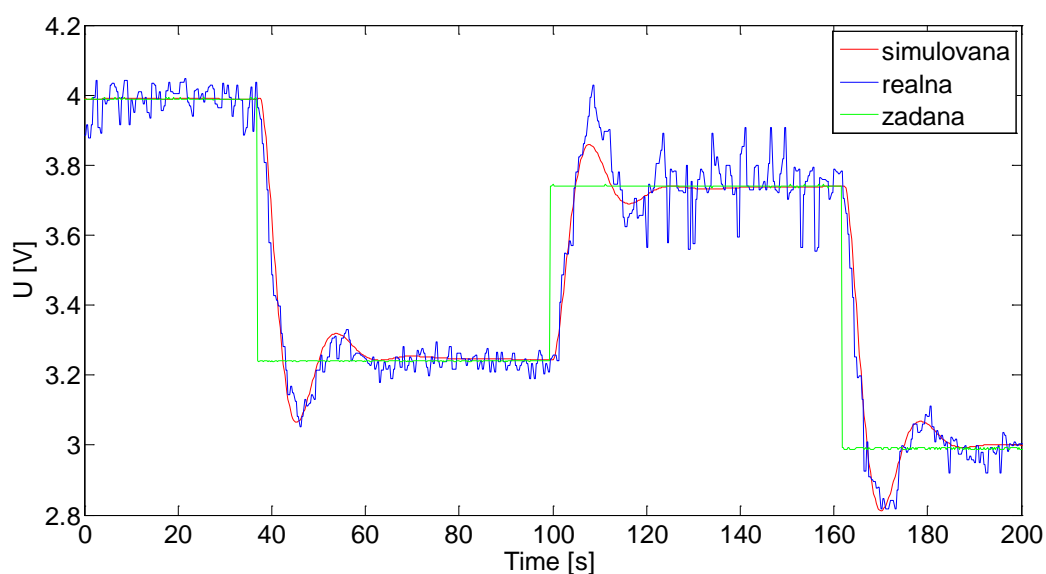
V obr. 3.8 je výstup RT In3, vstup z multifunkční karty, žádanou hodnotou, která byla zadaná na PLC. RT In4 je aktuální poloha letadla. Porovnával jsem simulovanou polohu letadla a reálnou. V grafech (Graf 6 a Graf 7) níže je modrou barvou reprezentován reálný proces a červenou simulace. V této chvíli bylo letadlo ovládáno pomocí PLC a počítač zde fungoval jen pro záznam dat, takže nezasahoval vůbec do řízení letadla.

Oblast testování byla určená pracovní oblastí, která je popsána v kapitole 2.1 Statické vlastnosti. Bylo potřeba tuto oblast dodržet, neboť parametry regulátoru byly nastaveny pro tuto pracovní oblast a regulace mimo ní se mohla lišit od simulované mnohem více než

v pracovní oblasti. Toto podezření se po vyzkoušení nepotvrdilo, viz kapitola 5.3 Aplikace a měření.



*Graf 13 – Porovnání regulace*



*Graf 14 – Porovnání regulace*

### 3.4 Vyhodnocení

Úkolem v této části bakalářské práce bylo se co nejvíce přiblížit regulací reálného systému k simulační regulaci. Podmínka, která byla splněna, bylo dodržení pracovní oblasti,

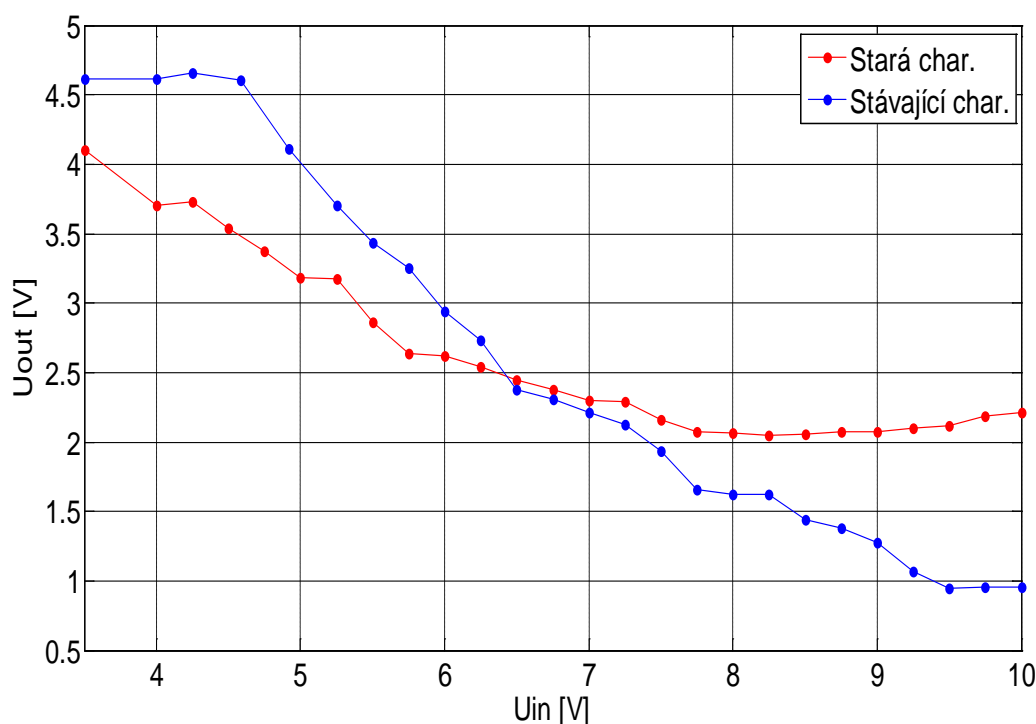
neboť se žádaná hodnota pohybovala od 2.75 V do 4.5 V. Skoky jsou voleny tak, aby nedocházelo k jejich opakování, což zvětšilo množství získaných dat.

Výstupem a kontrolou mé práce jsou grafy (12 a 13), kde můžeme sledovat výchylku letadla v reálném čase. V grafech jsou znázorněny žádané hodnoty simulované i reálné. Zelená křivka značí skokovou změnu žádané hodnoty v čase. Skoková funkce je způsobena zadáváním vyžádaných hodnot na grafickém panelu. Tímto se skokově změní žádaná hodnota. Na reálné soustavě letadla není tato změna možná. Díky identifikaci systému jsem mohl navrhnout a vyzkoušet parametry regulátoru v simulaci, což bylo o mnoho rychlejší, než odzkoušení parametrů PID regulátoru na reálném systému. Výsledkem tohoto procesu byla simulovaná červená funkce v grafech, která byla navrhnutá tak, aby byla doba náběhu co nejmenší. Zvolené parametry regulátoru způsobují překmit funkce. Při pokusech se změnami parametrů, jsem se jej pokoušel odstranit. To způsobilo, že doba ustálení výrazně narostla. Což byl důvod rozhodnutí ponechání náběhu s překmitem. Šumy – jemné zákmity jsou způsobeny provozem letadla - především vibracemi motoru. Při zvýšení úhlu výchylky a snížení napětí se regulace velmi blíží nasimulovaným hodnotám (modrá křivka). Při opačné činnosti – snižování úhlu, byla křivka více zatížena šumem.

Při vyhodnocení výsledku je nutné přihlédnout k šumovému rušení signálu. I přes zmíněné rušení jsou si křivky velice podobné, což svědčí o dobré implementaci řídicích algoritmů, a tudíž splnění zadaného bodu bakalářské práce.

## 4 PID s proměnným zesílením

Během práce bylo nutné předělat napájení modelu, čímž došlo k velkým změnám na statické a částečně i na dynamické charakteristice. Tento bod bakalářské práce byl do zadání zahrnut ještě v době, kdy statická charakteristika byla značně nelineární, viz graf 15. V současném stavu vykazuje charakteristika menší nelinearitu.



Graf 15 - Porovnání statických charakteristik

Na původní statické charakteristice (červená barva) je velice zřejmá nelinearita, kdy kolem 8 V na vstupu do modelu, začne hodnota výstupního napětí růst s rostoucí hodnotou vstupního napětí. Oproti tomu stávající charakteristika (modrá) má pořád stejnou tendenci klesat s hodnotou výstupního napětí s rostoucím vstupním napětím.

Aby byla zvýšena přesnost regulace, můžeme měnit parametry regulátoru přímo nebo pouze zesílením akčního zásahu. Při pouhém zesílení vytvoříme PID regulátor proměnným zesílením. Ke konstrukci jsem použil PID regulátor v paralelním tvaru, tím budu pouze měnit proporcionální složku regulátoru. Integrovaní i derivační zůstanou nadále stejné, viz [9].

Proporcionální složka regulátoru se bude měnit lineárně s aktuální hodnotou výstupu, v tomto případě aktuální polohou letadla.

## 4.1 Nelinearita

Nelineární procesy lze rozdělit do dvou kategorií. První jsou procesy, kde jsou změny předvídatelné. V těchto procesech existují procesy, či pomocné proměnné v jejich okolí (tyto proměnné se označují zkratkou SV, scheduling variable). Pomocí těchto proměnných lze měnit v programu parametry regulátoru, např. časovou konstantu či zesílení regulátoru. Jako příklad lze uvést: ovládaní nelineárního filtru, jehož statickou charakteristiku předem známe, nebo tepelného výměníku. Viz [8] PI regulace procesů s nelineárním chováním.

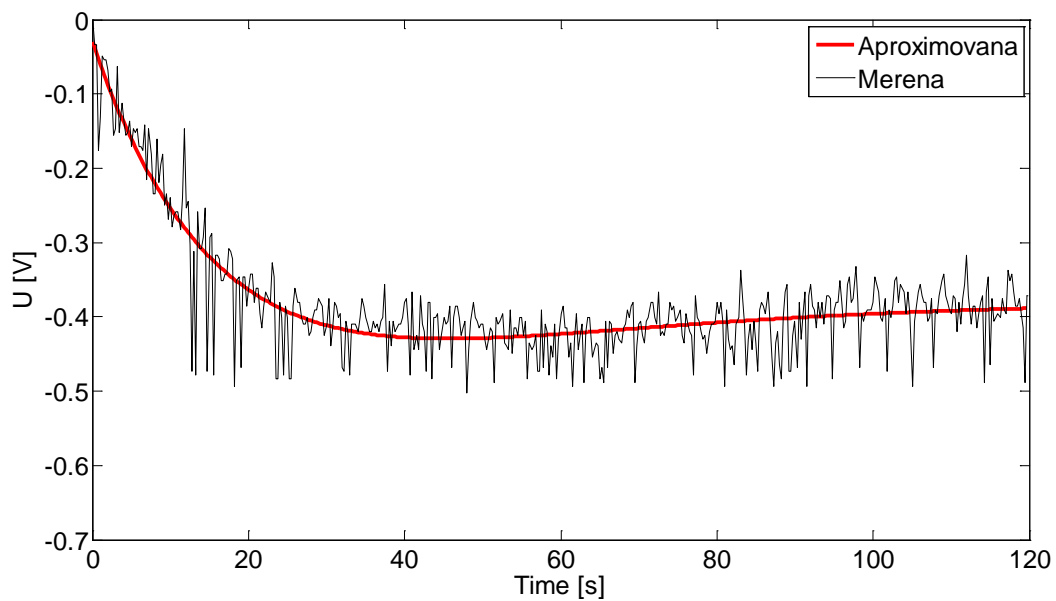
Cituji z [8]: „Pro regulaci těchto procesů lze použít adaptibilní regulátory. Ty si sami mění svoje parametry v závislosti na změně aktuálního chování systému. Je-li však změna příliš rychlá, může dojít k chybě v adaptaci, což způsobí značnou chybu regulace. U předem známých procesů lze naprogramovat adaptaci systému předem. Tím bude přizpůsobení mnohem rychlejší a šance, že se způsobí chyba vlivem špatné adaptace, se velice sníží.

Metoda gain scheduling představuje efektivní přístup k regulaci nestálého procesu s předpověditelnými změnami. Princip je jednoduchý, při změně časové konstanty se změní zesílení regulátoru.“

Do této kategorie patří regulace modelu letadla. Statická charakteristika je předem známá a chování lze doměřit, čímž se zvýší možnost adaptace systému a linearizace řízení. Rovnice popsané níže (4.5, 4.6 a 4.7) je možné považovat za scheduling variable. Přizpůsobují proporcionální složku PID regulátoru aktuální poloze letadla. Díky omezení modelu letadla, nelze vypadnout z adaptace, a tím nedojde k chybě v přizpůsobení.

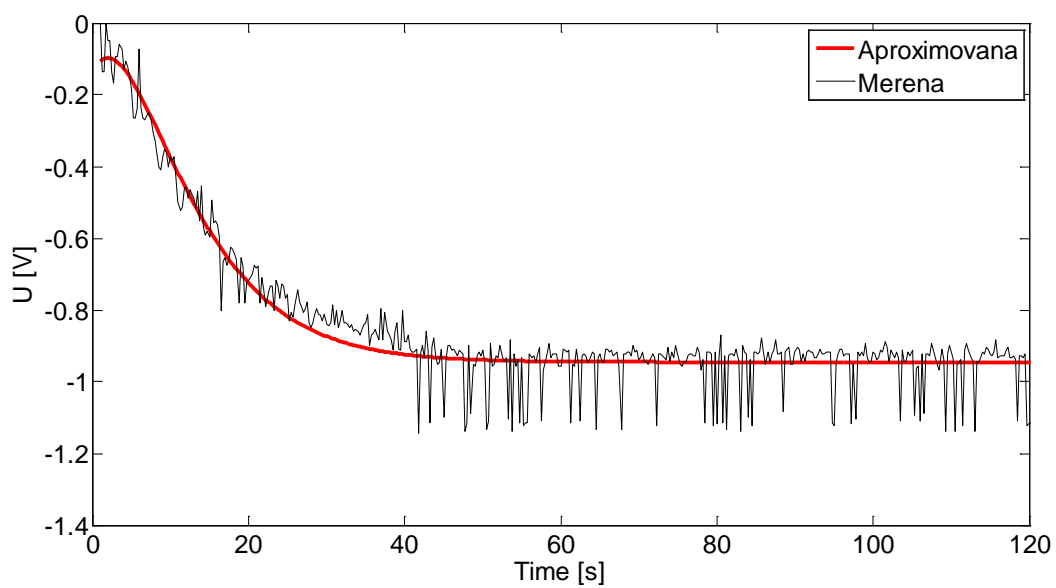
## 4.2 I a D koeficienty

Pro výpočet koeficientů bylo zapotřebí vybrat více pracovních bodů. Zvolil jsem pracovní body v 3.7 V, 3.1 V, 1.9 V a 1.24 V na výstupu modelu. Tyto hodnoty jsou zvoleny tak, aby bylo pokrytí změn v dynamických vlastnostech co největší. Přenosové funkce jsem zjišťoval stejným způsobem, který je popsán v kapitole 2.2 Dynamické vlastnosti. Tímto postupem jsem došel ke čtyřem přechodovým funkcím:



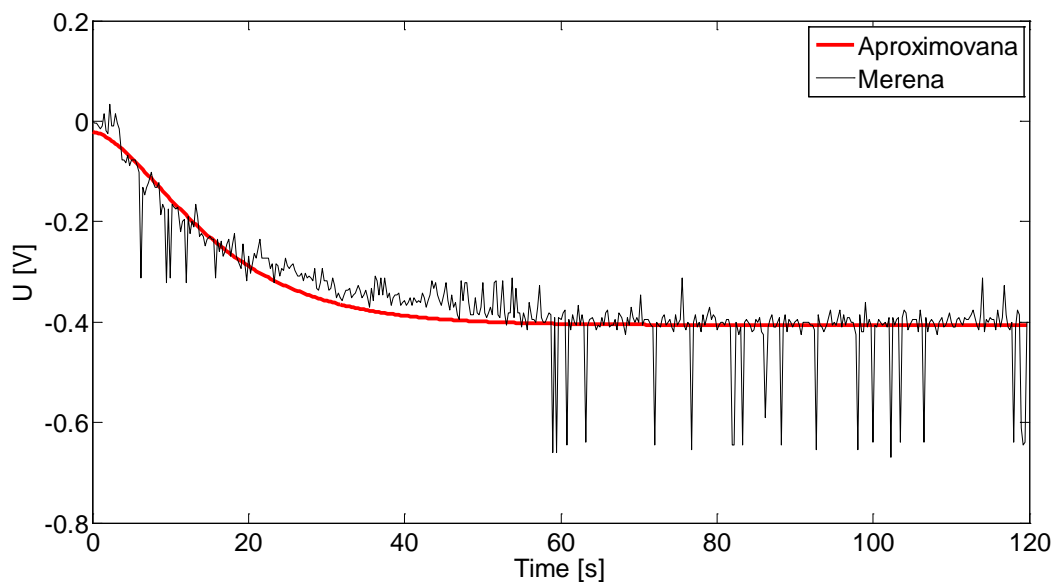
Graf 16 – Aproximovaná křivka vyjádřená rovnicí 4.1

$$F(s) = \frac{-0.8182}{20.3266s^2 + 9.017s + 1} \quad (4.1)$$



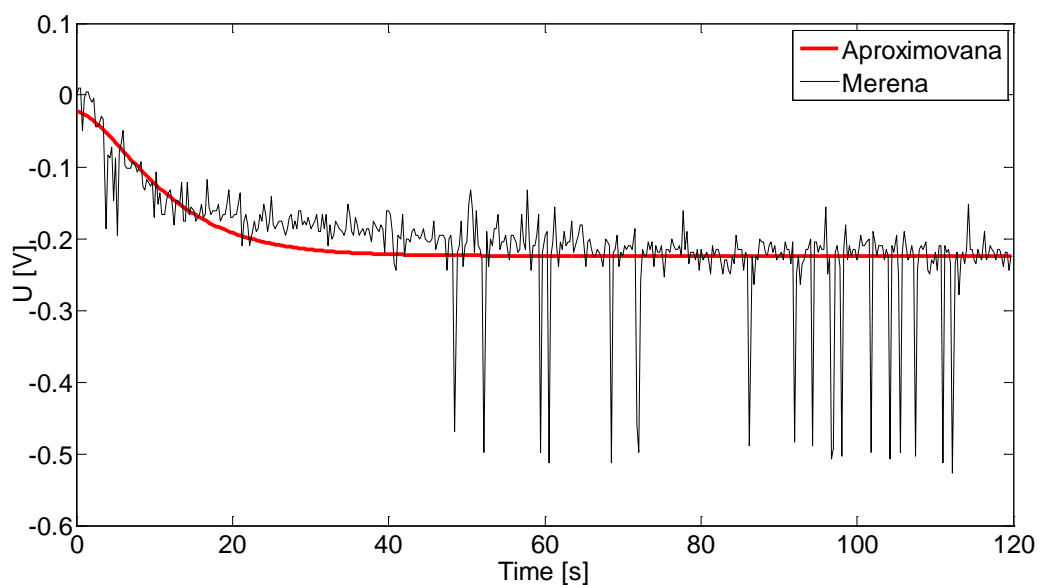
Graf 17 - Aproximovaná křivka vyjádřená rovnicí 4.2

$$F(s) = \frac{-0.9451}{46.5765s^2 + 13.6494s + 1} \quad (4.2)$$



Graf 18 - Aproximovaná křivka vyjádřená rovnicí 4.3

$$F(s) = \frac{-0.8121}{71.2961s^2 + 16.8874s + 1} \quad (4.3)$$



Graf 19 - Aproximovaná křivka vyjádřená rovnicí 4.3

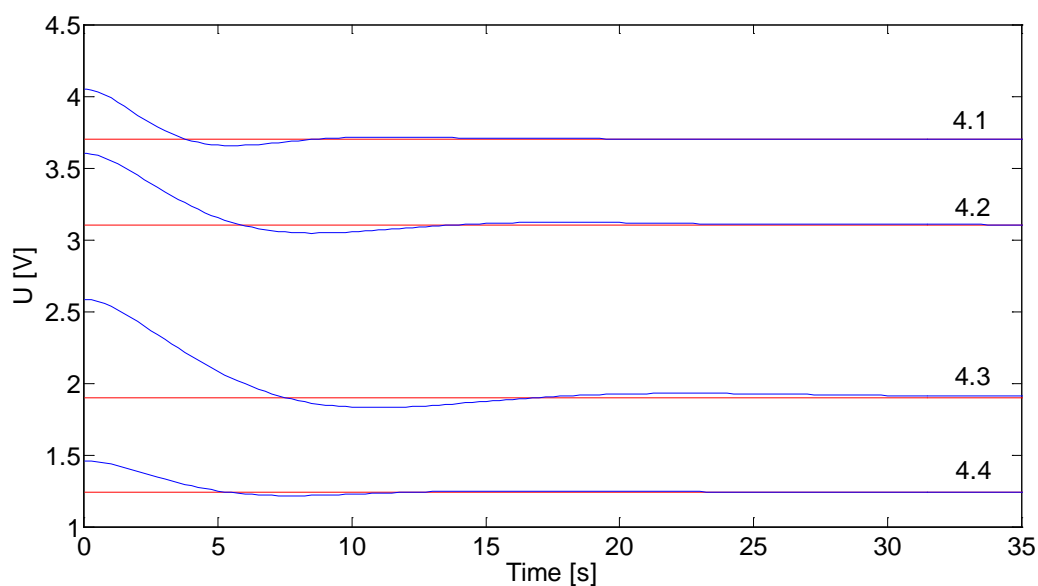
$$F(s) = \frac{-0.4482}{38.2381s^2 + 12.3674s + 1} \quad (4.4)$$

Dalším krokem byl výpočet koeficientů. Ten byl popsán v kapitole 3.1 Návrh regulátoru. Výsledné koeficienty:

Tabulka 5 – Výsledné koeficienty

	<b>r0</b>	<b>Ti</b>	<b>Td</b>
<b>4.1</b>	6.96	12.24	0.65
<b>4.2</b>	6.04	18.54	0.98
<b>4.3</b>	7.07	22.95	1.20
<b>4.4</b>	12.71	16.80	0.89

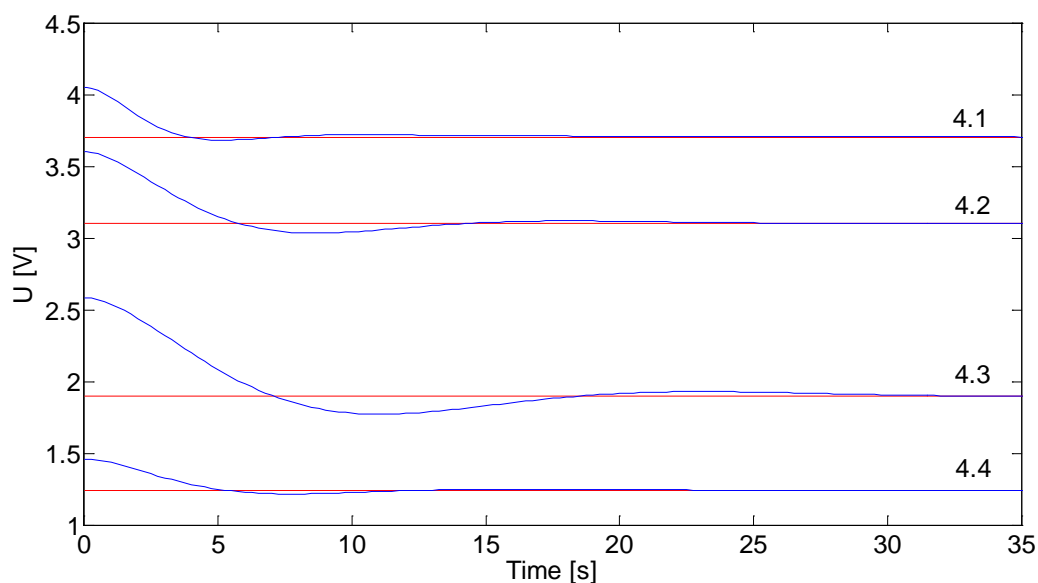
V grafech níže (graf 20 a 21) jsou zobrazeny všechny regulované funkce (4.1, 4.2, 4.3, 4.4) s parametry regulátorů, viz tab. 5. Modrou barvou jsou znázorněné simulované polohy letadla a červenou žádané hodnoty.



Graf 20 – Průběhy pro koeficienty z tab. 2

Tyto koeficienty bylo nutné upravit tak, aby se jednalo pouze o změnu proporcionální složky regulátoru. Koeficienty  $T_I$  a  $T_D$  musí být po celou dobu regulace stejné, neměnit se. Po provedení několika experimentů jsem zvolil  $T_I=16.39$  a  $T_D=0.92$ .





Graf 21 – Průběhy pro koeficienty  $T_I = 16.39$  a  $T_D = 0.92$

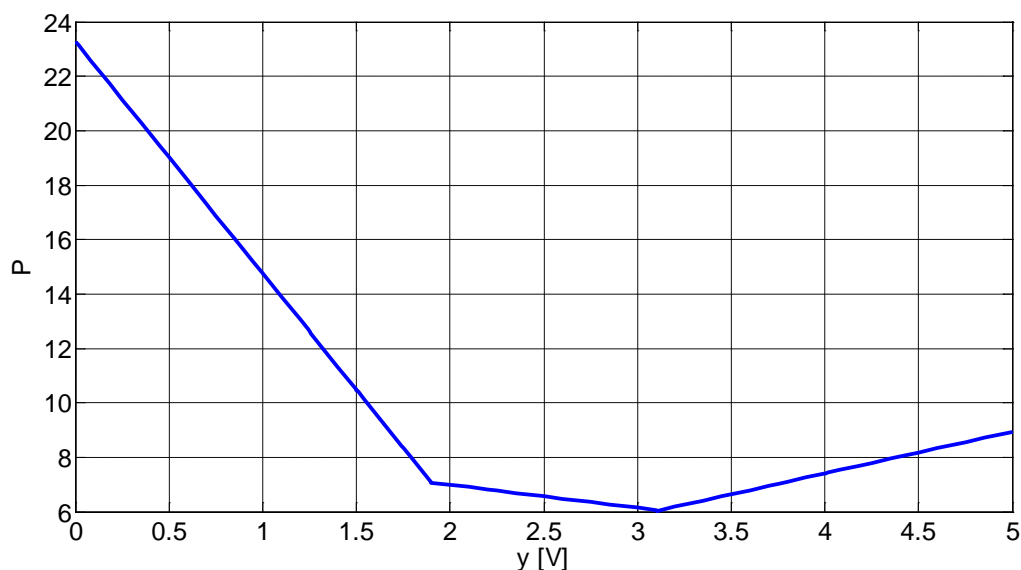
### 4.3 Aplikace a měření

Pro lineární změnu proporcionálního zesílení regulátoru bylo potřeba nalézt lineární funkci, která by mi tuto závislost popsala. Tyto lineární rovnice (4.5, 4.6, 4.7) byly nalezeny tak, že koncové body byly proloženy přímkou. Aproximace pomocí přímek byla snazší pro realizaci v programu, který byl implantován do PLC.

$$x \in \langle 3.1, 5 \rangle \quad y = 1.533x + 1.277 \quad (4.5)$$

$$x \in \langle 1.9, 3.1 \rangle \quad y = -0.85x + 8.69 \quad (4.6)$$

$$x \in \langle 1.9, 0 \rangle \quad y = -8.54x + 23.29 \quad (4.7)$$



Graf 22 – Výpočet P

```

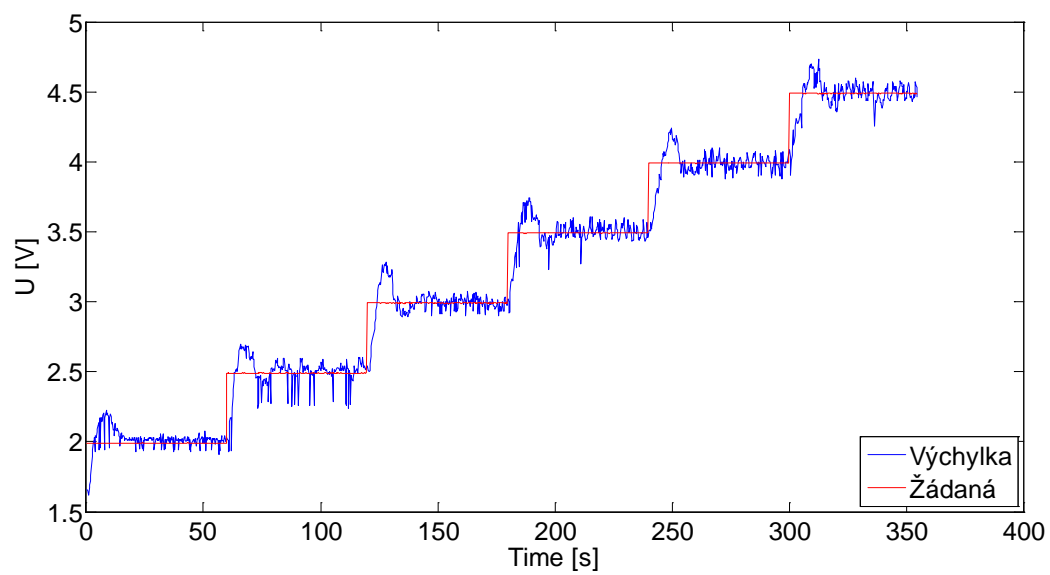
IF (vyber=3) THEN (*vlastni P*)
  IF ((reg_y>3.7) OR ((reg_y<=3.7)AND(reg_y>3.1))) THEN
    reg_P:=1.533*reg_y+1.277;
  END_IF;
  IF ((reg_y<=3.1)AND(reg_y>1.9)) THEN
    reg_P:=-0.86*reg_y+8.69;
  END_IF;
  IF ((reg_y<=1.9)OR((reg_y<=1.24)AND(reg_y>1.24))) THEN
    reg_P:=-8.54*reg_y+23.29;
  END_IF;
END_IF;

```

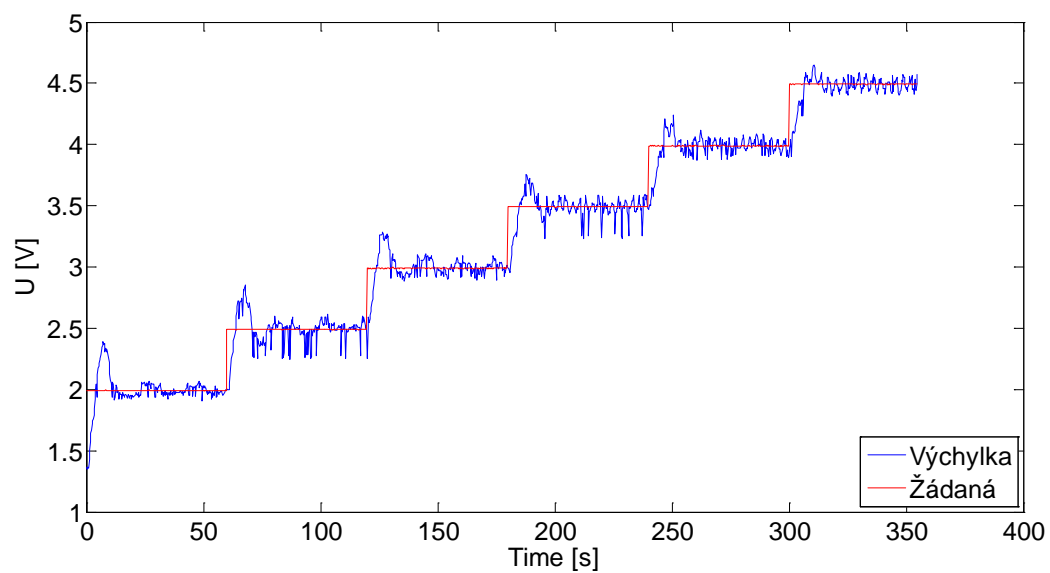
Obr. 5.1 – Změna P složky v programu

Rovnice pro výpočet změny proporcionální složky (4.5, 4.6 , 4.7) byly přepsány do programu GX Developer, viz Obr. 4.1., a implementovány do PLC

Provedl jsem měření jak s algoritmem se změnou proporcionální složky, tak i bez ní. Z grafů uvedených níže vyplývá, že regulace výstupu je velice podobná, takřka totožná, viz Graf 12 a 13. Důvodem může být změna statických vlastností modelu, která je popsána na začátku kapitoly.



Graf 23 – Záznam polohy bez algoritmu pro změnu  $P$  složky



Graf 24 – Záznam polohy s algoritmem pro změnu  $P$  složky

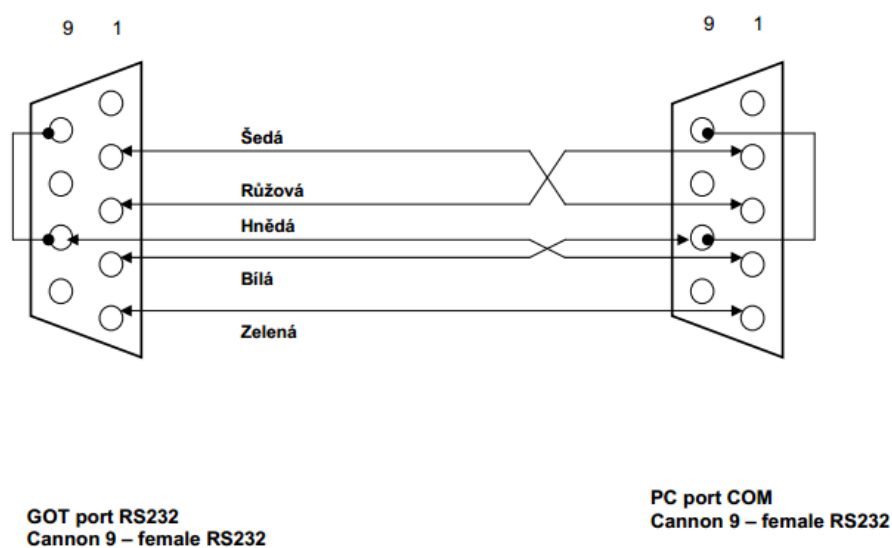
## 5 Grafický panel

Pro programování grafického panelu je nutno nainstalovat softwarový balík GT Designer 2. Po spuštění nového projektu je možné zadat typ panelu a nastavení barev. V mém případě jsem nastavil typ panelu GT15 – V(640x480). Nastavení barev jsem ponechal standardní. Poté je nutné zvolit typ controlleru - Melsec Q.

### 5.1 Komunikace

Po inicializaci je potřeba nastavit komunikaci mezi PC a panel. Možnosti jsou dvě: USB a RS 232, viz [4]. Pro komunikaci po USB je nutné doinstalovat ovladače, které nebyly součástí balíku a ani se nenacházely na dodávaném CD-ROMU. Mně tato komunikace po instalaci ovladačů nefungovala. Sice můj operační systém hlásil připojení panelu, ale také chybu na straně driveru. Proto jsem zvolil komunikaci po RS 232

Pro komunikaci po sériové lince je potřeba zapojit konektory přesně podle zobrazení:



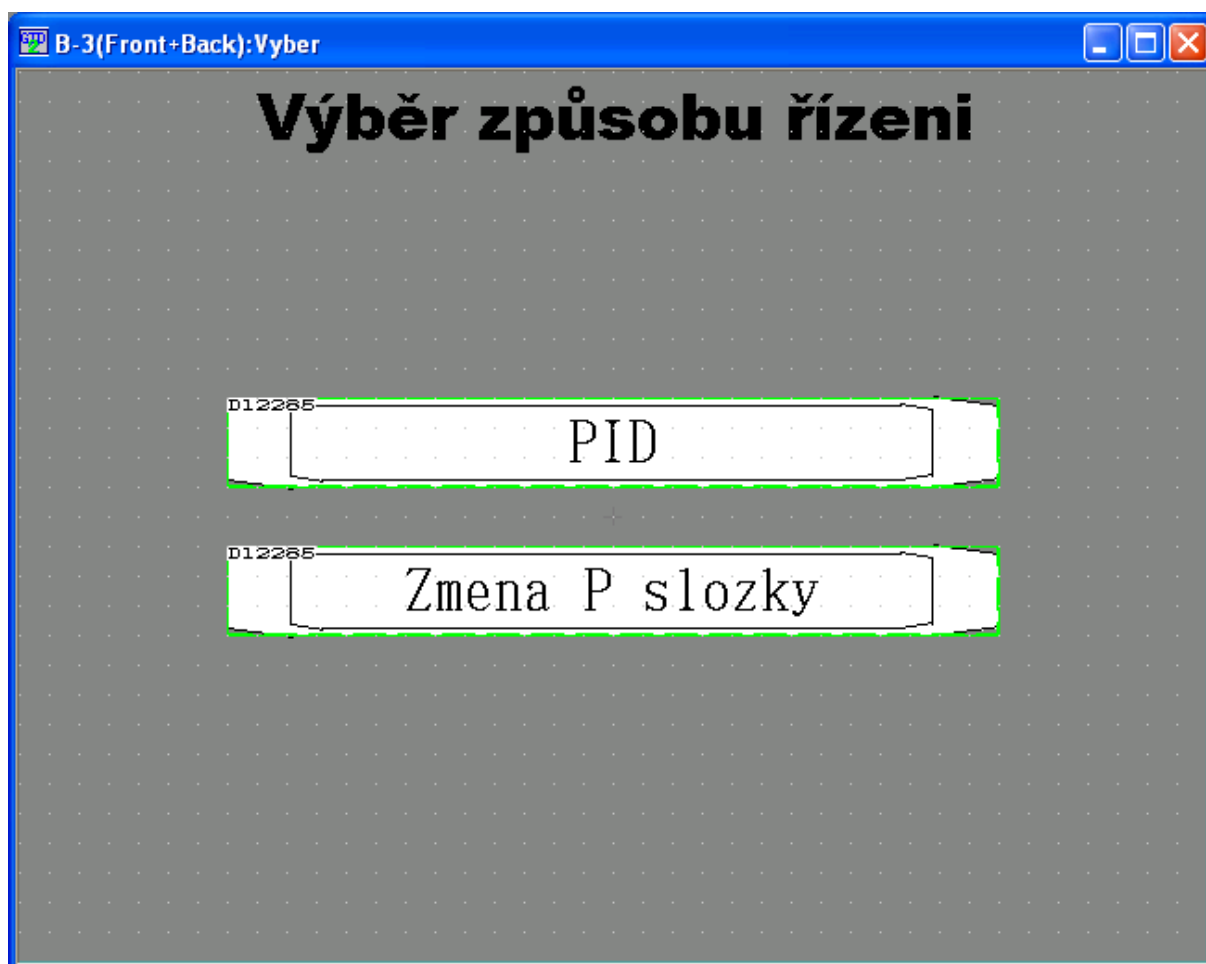
Obr. 6.1 – Zapojení RS 232

V levém horním rohu obrazovky je skryté tlačítko, které spustí nastavení hlavního menu panelu. Zde je nutné v komunikaci nastavit ChNo. 9 na Host(PC) a ChNo. 1 na Q Bus.

## 5.2 Vytváření uživatelského programu pro ovládání dané úlohy

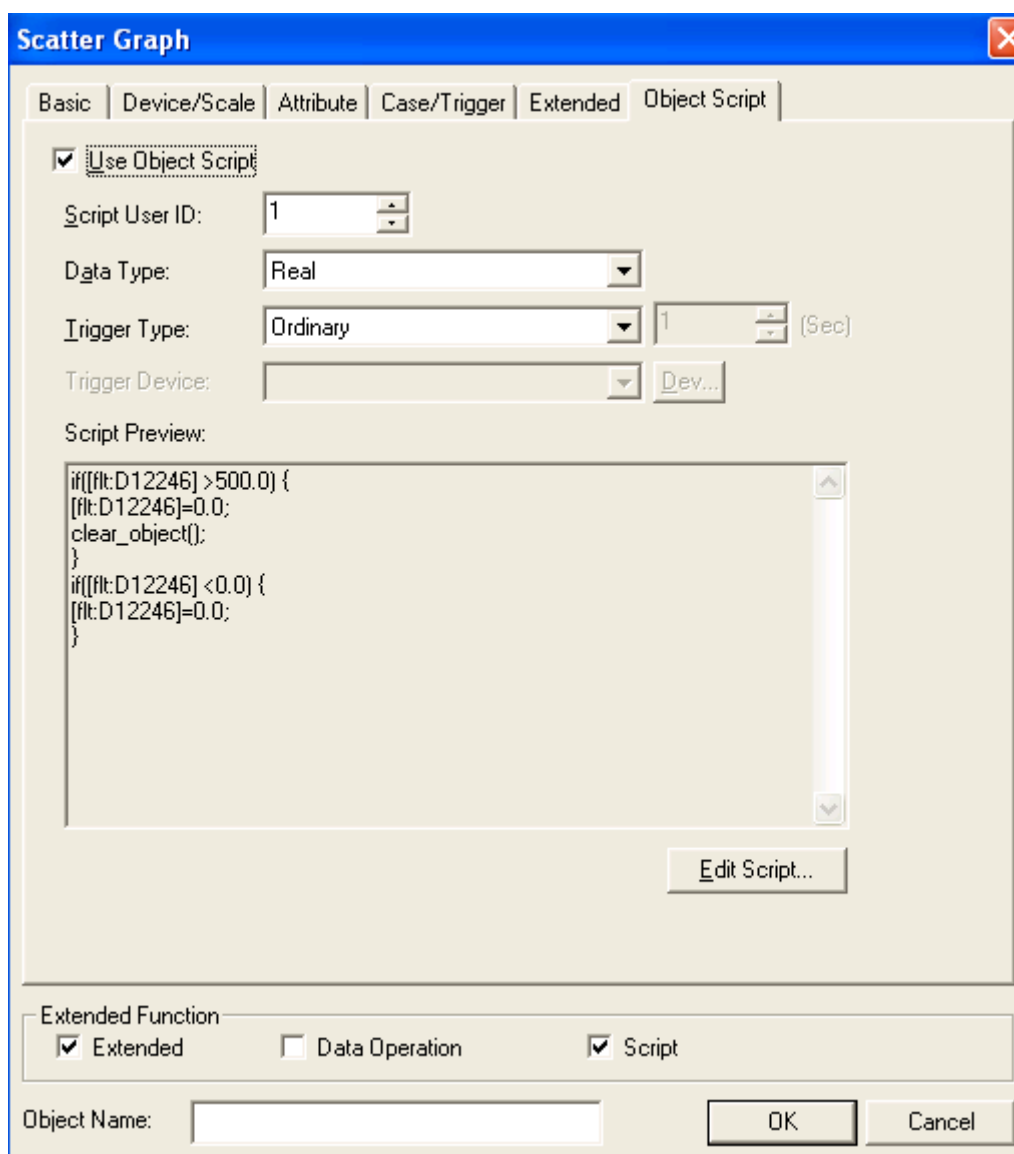
Uživatelský program se vytváří pomocí GT Designer 2. Celý proces návrhu se skládá nejprve z přidávání Base Screenů, což jsou podkladové obrazovky, na které poté se instalují další ovládací prvky. Mezi ně patří Multi Action Switch, u něhož lze měnit více parametrů najednou. Například číslo Base Screenu, který bude aktuálně zobrazen, může se měnit i hodnota proměnné v paměti kontroléru, atd.

Při návrhu designu uživatelského programu jsem se inspiroval DP Hlavatého [11].



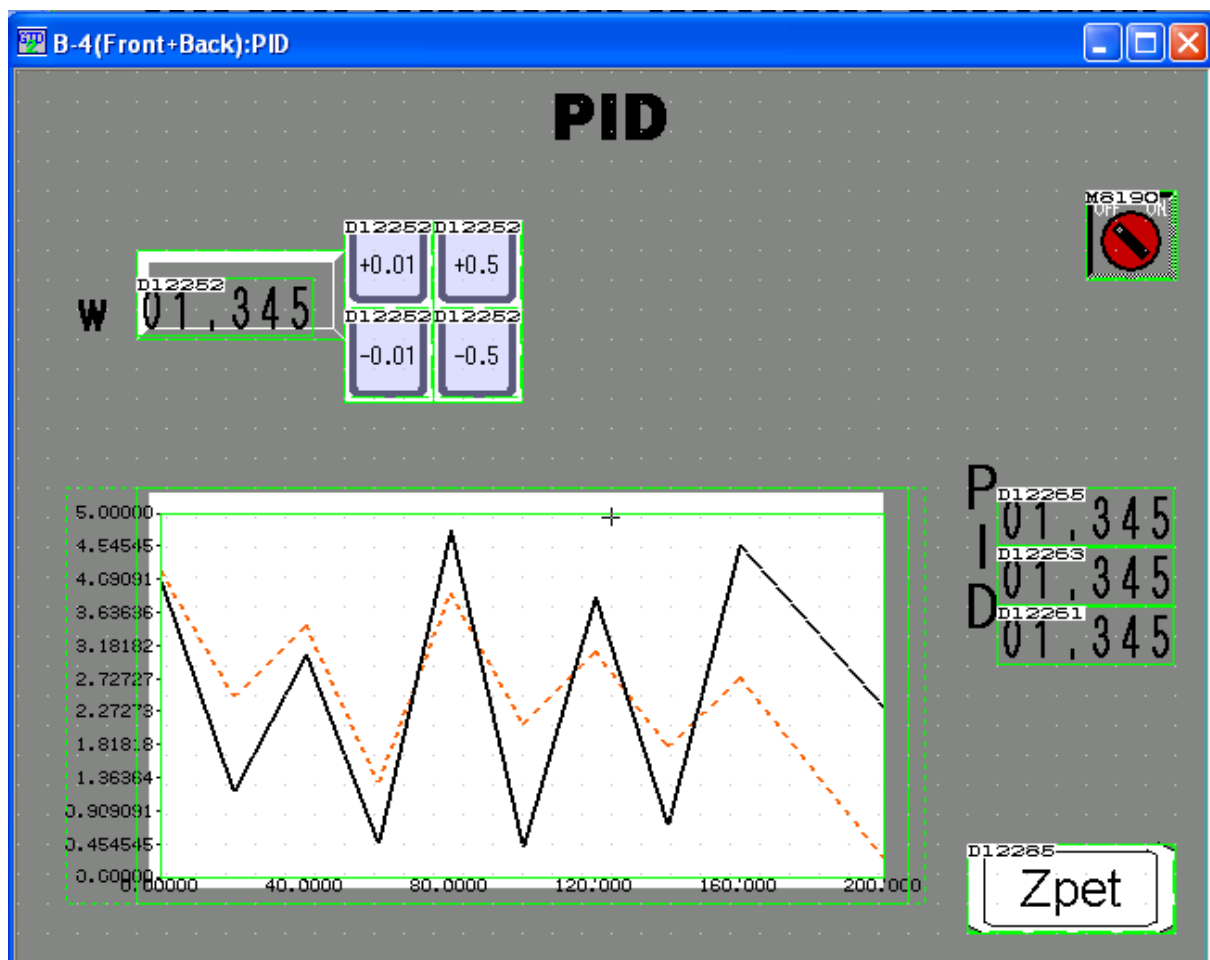
Obr. 6.2 – Hlavní okno

Objekty jsou přímo spojeny s proměnnými pomocí adres, kterou lze zjistit u proměnných v programu GX Devoper. U každého objektu je možné dopsat skript, viz obr. 6.3, který se vykonává v panelu nikoliv v PLC.



Obr. 6.3 – Ukázka skriptu – Scatter Graph, resekce zobrazení grafu

Při volbě regulace se nahrají předvolené koeficienty regulátoru, které mohou být změněny.



Obr. 6.4 – Ukázka Base okna

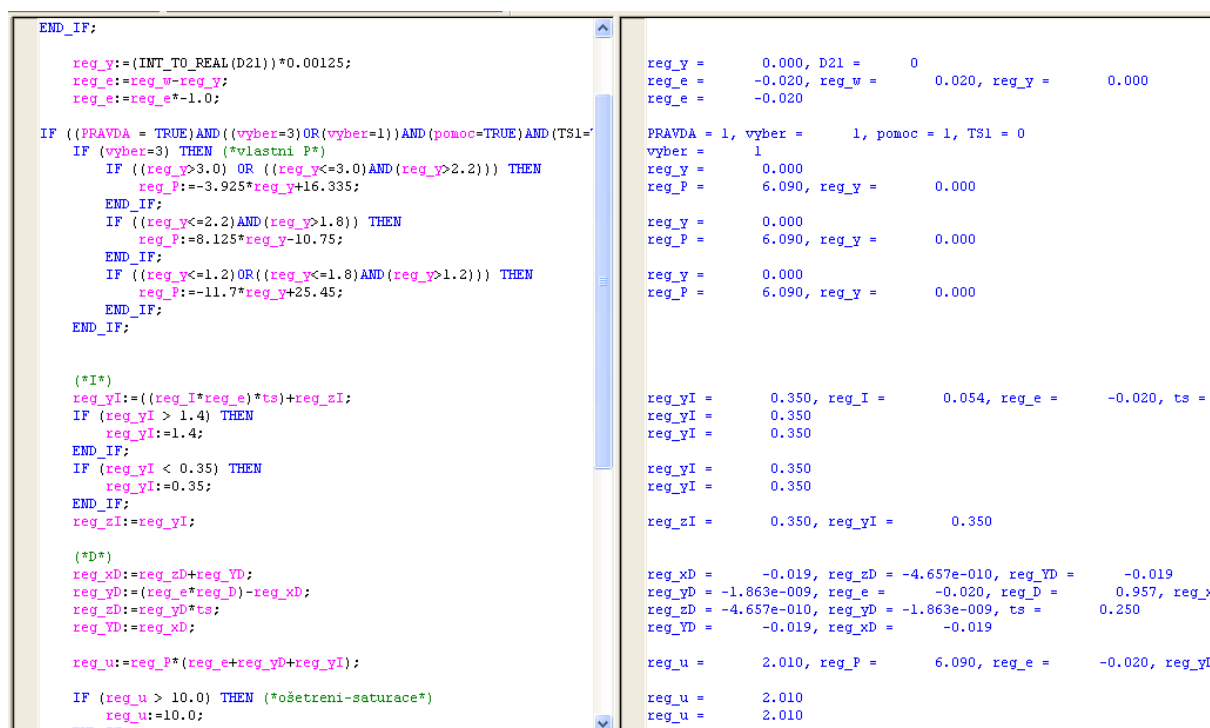
Proměnné W, P, I a D jsem řešil jako „Numerical Input“. Ten umožňuje hodnoty měnit pomocí číslíkové klávesnice. Žádanou hodnotu lze rychle změnit za pomoci čtyř tlačítek umístěných vedle hodnoty a omezil jsem jí za pomoci skriptů na hodnoty mezi 0 a 5. Interní časovač, který jsem inicializoval v hlavním programu, měl na starosti přesné časování grafu.

Regulace se spustí za pomoci polohového přepínače v pravém horním rohu. Objekty jsem volil tak, aby bylo ovládání panelu snadné.

## 6 Online monitoring a export dat do PC

Pro export dat do PC je nutno nainstalovat software MX Component. Tento program je dodáván s GX Developer, ale pouze v shareware verzi, která je časově omezená. Mně se jej nepodařilo nainstalovat. MX Component by měl projít PLC s uživatelským prostředím v programu Microsoft Excel. Kvůli chybě při instalaci jsem tuto možnost nemohl ověřit. Výrobce doporučuje operační systém Win. XP. Já jsem se tuto podmínku snažil splnit alespoň tím, že jsem si nainstaloval virtuální počítač s operačním systémem Win. XP.

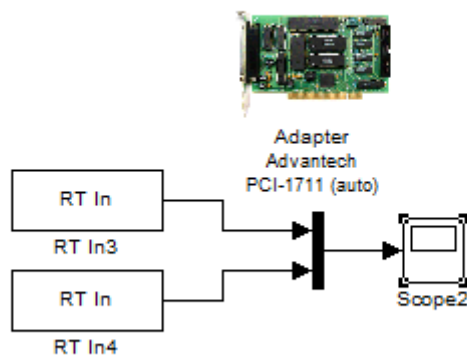
Online monitoring můžeme zajistit v programu GX Developer, který umožňuje nahlížet do paměti PLC při chodu programu, viz Obr. 6.1. V levé části okna se nachází kód, jenž byl implementován do PLC. V pravé části jsou uvedeny aktuální hodnoty proměnných s pozicí stejnou jako v algoritmu. Tento styl je velice přehledný a do jisté míry pomáhá s úpravami a laděním.



Obr. 7.1 – Monitoring

Bylo potřeba zajistit záznam dat jinou metodou. Zvolil jsem způsob záznamu dat pomocí programu *MATLAB* a speciální karty. K záznamům jsem vytvořil velmi jednoduché schéma, viz Obr. 6.1, které jsem v případě potřeby rozšiřoval o více vstupů.





*Obr. 7.2 – Schéma pro záznam dat*

V nastavení vstupů jsem volil rozsah od 0 do 10 V s vzorkovací periodou 0.25 s. Ve scope bylo potřeba potvrdit záznam dat, zrušit omezení počtu vzorků a pojmenovat proměnou. Data se ukládali do workspace do proměnné, jako pole s indexem.

## Závěr

Při měření statické charakteristiky modelu jsem našel značně lineární části charakteristiky. V těchto oblastech jsem prováděl měření dynamických vlastností. Nalezená převodní charakteristika byla druhého řádu bez dopravního zpoždění.

Při návrhu PID regulátoru jsem musel počítat se šumem, který vzniká při provozu modelu letadla. Bylo nutné upravit derivační složku regulátoru, aby šum nezasahoval do výpočtů akčního zásahu. Při tvorbě řídicích algoritmů jsem vycházel z navrhnutého simulovaného PID regulátoru. Pro ověření realizace PID regulátoru jsem porovnával simulovanou a reálnou polohou letadla a rozdíl mi vyšel minimální, téměř nulový.

Při aplikaci PID s proměnným zesílením se anuluje nelinearita systému. Z důvodu změny napájení modelu během práce se statická charakteristika výrazně přiblížila lineární charakteristice. Tento fakt se projevil při porovnání jednoduchého PID regulátoru s regulátorem s proměnným zesílením. Oba výstupu jsou totiž téměř shodné.

Při realizaci ovládání prostřednictvím grafického dotykového panelu jsem se snažil o jednoduchost a návaznost jednotlivých prvků.

Pro nahlížení do paměti regulátoru jsem zprovoznil vnitřní funkci programu GX Developeru.

Snažil jsem se, co nejlépe dostat všem zadaným bodům bakalářské práce. Zadání jsem splnil, i přes nastalé komplikace s napájením modelu.

## Seznam použité literatury

- [1] HLAVA, J. Prostředky automatizačního řízení II. Praha: ČVUT, 2000. 162s.  
Dostupné z WWW: < [http://www.fm.tul.cz/~jaroslav.hlava/par/Skripta\\_PAR.pdf](http://www.fm.tul.cz/~jaroslav.hlava/par/Skripta_PAR.pdf) >
- [2] *Lombaplc* [online]. 2010 [cit. 2012-05-10]. MELSEC System Q:Digital-Analog Converter Modules Q62DA, Q64DA, Q68DAV/DAI GX Configurator-DA. Dostupné z WWW: <[http://lombaplc.tf.itb.ac.id/wp-content/uploads/2010/02/Q\\_DIGITAL\\_ANALOG\\_CONVERTER\\_.PDF](http://lombaplc.tf.itb.ac.id/wp-content/uploads/2010/02/Q_DIGITAL_ANALOG_CONVERTER_.PDF)>.
- [3] *Meltrade* [online]. 2010 [cit. 2012-05-10]. MELSEC System Q:User's Manual Q64AD ,Q68ADV/ADI, GX ConfiguratorAD . Dostupné z WWW: <[http://www.meltrade.hu/dokumentumok/1.\\_Programozhat%C3%B3\\_logikai\\_vez%C3%A9rl%C5%91k\\_\(PLC\)/1.4.\\_Modul%C3%A1ris\\_PLC\\_-\\_MELSEC\\_Q/1.4.3.\\_Anal%C3%B3g\\_%C3%A9s\\_h%C5%91m%C3%A9rs%C3%A9kletszab%C3%A1lyoz%C3%B3\\_modulok/1.\\_Anal%C3%B3g\\_bemeneti\\_modulok/Q62AD-DGH,Q64AD-GH\\_-\\_User's\\_Manual\\_SH\(NA\)-080277-K\\_\(01.08\).pdf](http://www.meltrade.hu/dokumentumok/1._Programozhat%C3%B3_logikai_vez%C3%A9rl%C5%91k_(PLC)/1.4._Modul%C3%A1ris_PLC_-_MELSEC_Q/1.4.3._Anal%C3%B3g_%C3%A9s_h%C5%91m%C3%A9rs%C3%A9kletszab%C3%A1lyoz%C3%B3_modulok/1._Anal%C3%B3g_bemeneti_modulok/Q62AD-DGH,Q64AD-GH_-_User's_Manual_SH(NA)-080277-K_(01.08).pdf)>.
- [4] *Meltrade* [online]. 2010 [cit. 2012-05-10]. GOT1000 GT15 User's Manual. Dostupné z WWW: <<http://f.ipc2u.pl/files/add/doc/174791/GT15%20Users%20Manual.pdf>>.
- [5] Scantime [online]. 2010 [cit. 2012-05-10]. MELSOFT-Software,GX Developer Version 8 Operating Manual. Dostupné z WWW: <[http://www.scantime.co.uk/\\_docs/Mi/Gx%20Developer%20op%20manual.pdf](http://www.scantime.co.uk/_docs/Mi/Gx%20Developer%20op%20manual.pdf)>.
- [6] Scantime [online]. 2010 [cit. 2012-05-10]. GX Developer Version 8 . Dostupné z WWW:[http://www.scantime.co.uk/\\_docs/Mi/Gx%20Developer%20op%20manual.pdf](http://www.scantime.co.uk/_docs/Mi/Gx%20Developer%20op%20manual.pdf)
- [7] PI regulace procesů s nelineárním chováním. PI regulace procesů s nelineárním chováním [online]. 2005 [cit. 2012-05-10]. Dostupné z WWW: [http://www.odbornecasopisy.cz/index.php?id\\_document=30500](http://www.odbornecasopisy.cz/index.php?id_document=30500)
- [8] Základy automatizace. 2007. Dostupné z: [http://www.elearn.vsb.cz/archivcd/FS/Zaut/Skripta\\_text.pdf](http://www.elearn.vsb.cz/archivcd/FS/Zaut/Skripta_text.pdf)
- [9] TOMÁŠ, FencI. ALGORITMUS VÍCEROZMĚROVÉHO REGULÁTORU. 2005. Dostupné z: [https://support.dce.felk.cvut.cz/mediawiki/images/4/4c/Dp\\_2005\\_fencI\\_tomas.pdf](https://support.dce.felk.cvut.cz/mediawiki/images/4/4c/Dp_2005_fencI_tomas.pdf)
- [10] Product Documentation for Matlab [online]. 2012 [cit. 2012-05-10]. Dostupné z: <http://www.mathworks.com/help/techdoc/index.html>
- [11] HUBKA, Lukáš. Aplikace programovatelného řídicích systémů pro řízení spojitých systémů. 2010.

## Příloha A – foto letadla

